Unidad 4 / Escenario 7

Lectura fundamental

Los enrutadores

Contenido

Palabras clave: capa de red, OSI, IPv4, IPv6, VLSM.

La capa de red es una de las más complicadas del sistema OSI, sin embargo vamos a hacer énfasis y distinción entre dos funciones primordiales que realiza esta, el reenvío (forwarding) y el enrutamiento (routing). El reenvío es la transferencia de un paquete de una interface a otra del mismo enrutador (router), por otra parte, el enrutamiento hace parte de toda la red en la cual está involucrado el enrutador, desde que viaja de su salida hasta la llegada del paquete de datos.

Vamos a ver cómo funciona la capa de red en el tránsito de datos de un punto a otro.

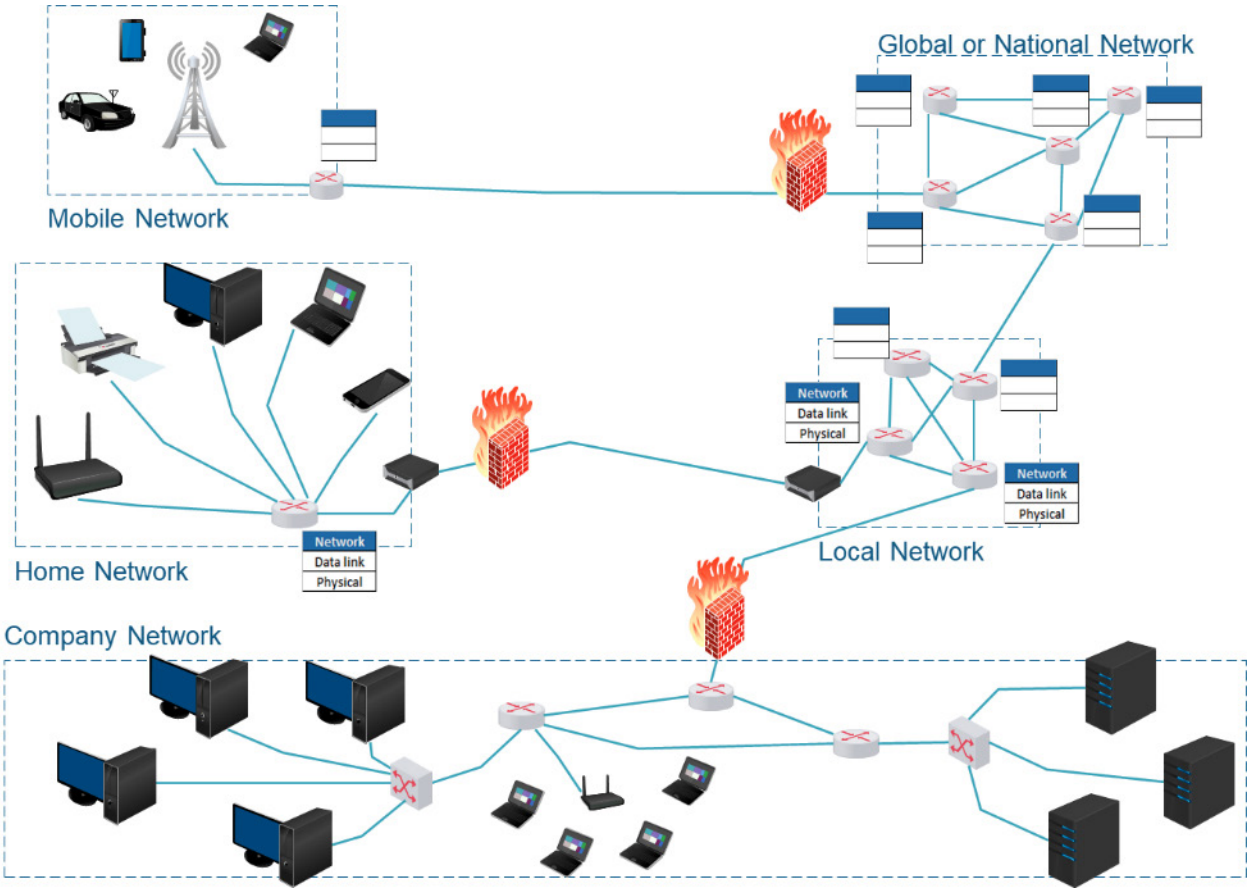


Figura 1. Tránsito de datos de punto a punto

Fuente: elaboración propia. Modificado de Kurose (2009)

La capa 3 (Red) en los host terminales, recibe los servicios de la capa 2 (enlace), voltajes, bits y tramas, con corrección de errores para des-encapsularlos, analizando el protocolo de red y entregar a la capa 4 (Transporte), para que esta haga el análisis en TCP o UDP y finalmente se entregue a la capa de aplicación.

Como se ve en la figura 1, los paquetes de datos van desde H1 hasta H2 pasando por la gran red de equipos, routers y enlaces que la forman, y el enrutamiento se crea desde que el paquete de datos sale del primer enrutador hasta que llega al último sin importar cuantos saltos hace. Cuando hablamos de saltos nos referimos al número de veces que pasa un dato de router a router, más adelante lo veremos más claro cuando estudiemos protocolos de enrutamiento.

1. Estructura interna de un enrutador

Un enrutador recibe los datos por un puerto físico y este cumple diferentes trabajos desde hacer todas las comprobaciones de la capa física, hasta realizar el nuevo entramado de conmutación y entregar por el puerto de salida al siguiente enrutador o equipo que reciba el paquete de datos.

Lo veremos más claro en la siguiente figura.

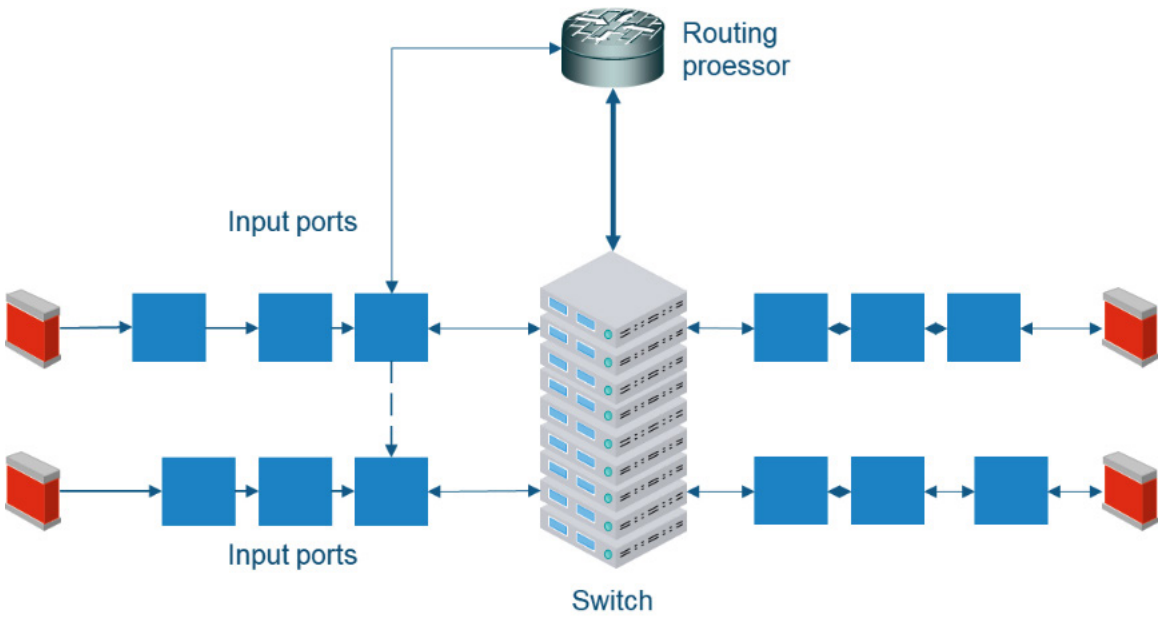


Figura 2. Estructura de un enrutador

Fuente: elaboración propia. Modificado de Kurose (2009)

Sin embargo, estas no son las únicas funciones del enrutador, pues este también debe saber hacia dónde va a enviar el paquete de datos. Como lo podemos ver la parte central del equipo es el entramado de conmutación (Switch fabric), y el procesador de enrutamiento (routing processor) es el que indica el camino por donde se debe ir. Aclarando el tema cuando hablamos de puertos, nos referimos a los puertos físicos del equipo, pues los puertos lógicos de software como los sockets van en las siguientes capas.

Los puertos bidireccionales funcionan tanto de entrada como de salida y tienen el siguiente comportamiento que se ve en la figura 3.

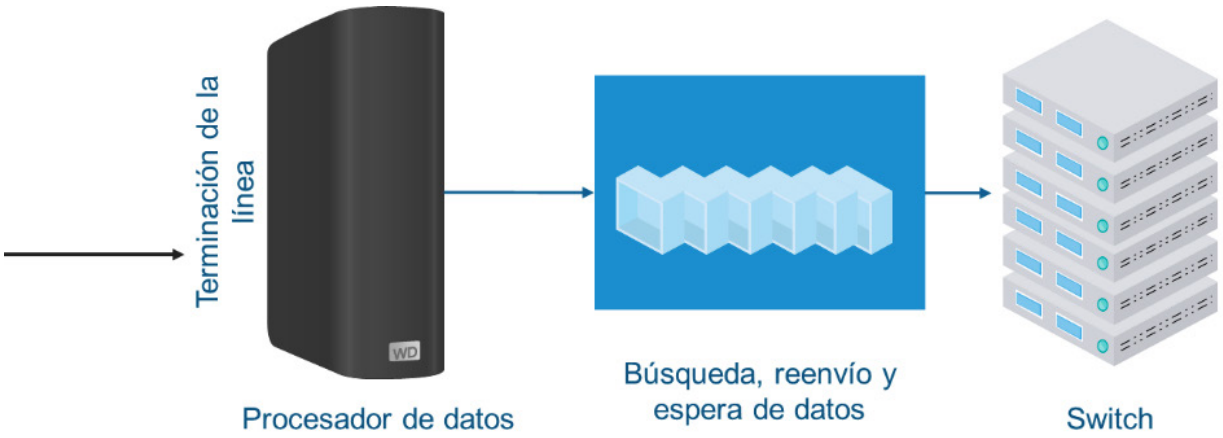


Figura 3. Puertos bilaterales

Fuente: elaboración propia. Modificado de Kurose (2009)

Como lo hemos nombrado antes esta terminación de línea y el procesamiento de las capas, física y de enlace se ven asociadas a las funciones de cada puerto, el módulo de búsqueda y reenvío del router es primordial en esta etapa, en muchos enrutadores aquí es donde se realiza esta toma de decisión, pues los puertos tienen una copia de la tabla de reenvío y no dependen del entramado de conmutación (Switch Fabric). Aunque esta tabla es hecha por el procesador de enrutamiento, se guarda una copia en cada puerto de entrada y hacer el reenvío mucho más rápido. Estos envíos aseguran que no haya encolamiento de paquetes por procesamiento de maquina ni estancamiento de memoria. El entramado de conmutación puede hacerse de varias maneras.

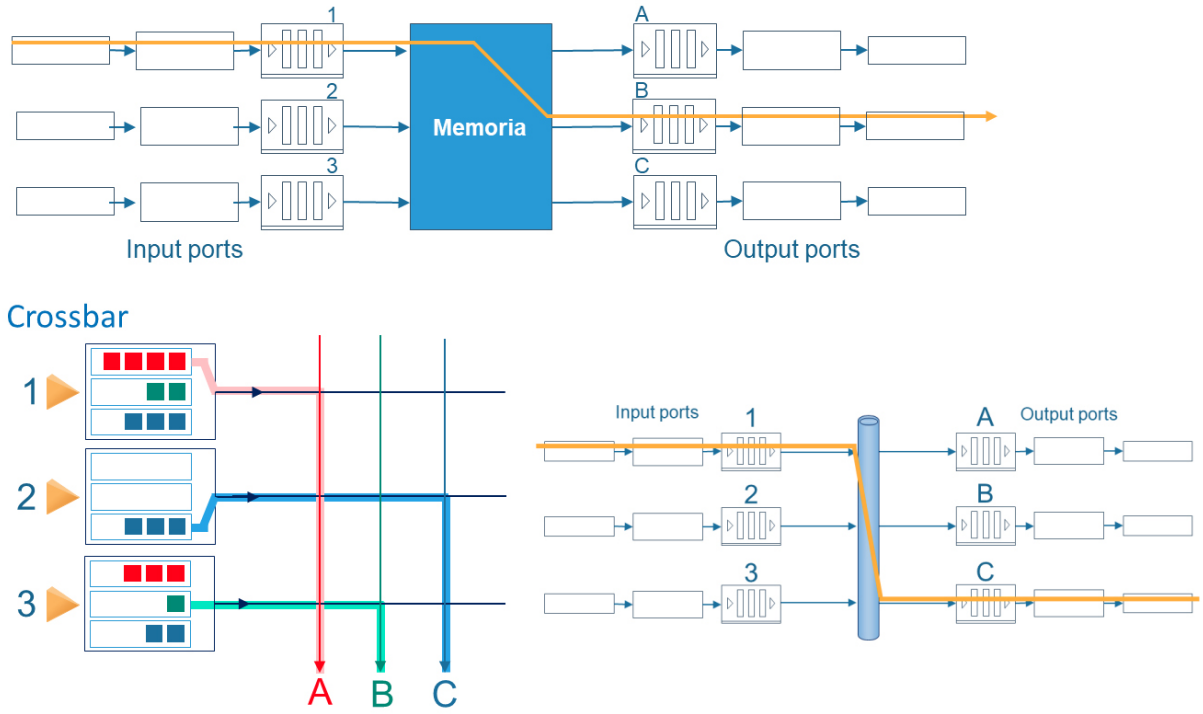


Figura 4. Entramado de conmutación

Fuente: elaboración propia. Modificado de Kurose (2009)

1.1. Conmutación por memoria.

En un principio los router eran maquinas muy simples, por decirlo de otra manera eran computadores de primera generación, donde usaban la memoria RAM y procesador para realizar las tareas básicas y por otra parte sus periféricos de entrada y salida eran los puertos de comunicaciones. Al llegar el paquete de datos al puerto, este era copiado a la memoria del procesador y allí se extraía la información de origen y destino, luego se copiaba en el buffer de salida sabiendo su destino y se buscaba en la tabla de enrutamiento y se colocaba en el puerto para ser entregado. Cabe denotar que la velocidad de procesamiento dependía de la memoria del enrutador y se media de esta manera, MEM/2.

1.2. Conmutación vía bus

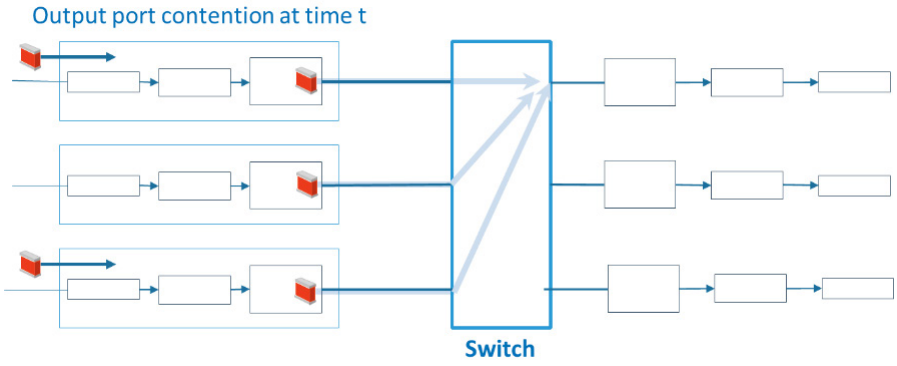
De esta manera el paquete que llega al puerto es transferido inmediatamente al puerto de salida por un bus compartido, sin la intervención del procesador de enrutamiento, sin embargo, nótese que en la memoria es igual, con la diferencia que tiene que pasar por ella, entrar y salir. Esta conmutación está restringida por la velocidad del bus.

1.3. Conmutación vía una red de interconexión

Esta conmutación pensó en mejorar a la anterior, con una red más moderna que nos diera más velocidad en relación a la del bus con una malla de buses como la que es usada en los multiprocesadores de computadores de alto rendimiento. Como se ve en la figura 4, un paquete llega y viaja a través del bus horizontal hasta encontrar el bus vertical que lo dirige al puerto de salida deseado, claro está sí este se encuentra libre, de lo contrario quedara bloqueado y en cola.

1.4. ¿Dónde se crea la congestión?

Revisando las funcionalidades de los enrutadores, los puertos de entrada y salida, la velocidad de procesamiento y los buffer de memoria, es lógico decir que si uno de ellos no tiene la capacidad necesaria de respuesta o si el tráfico es mayor a su velocidad de rendimiento, se va a llegar al congestionamiento de paquetes de datos y esto generara perdida de paquetes, como lo hemos hablado en escenarios anteriores la perdida de paquetes se debe al descarte de los mismos por el router. En la siguiente figura vemos el resultado de estas suposiciones.



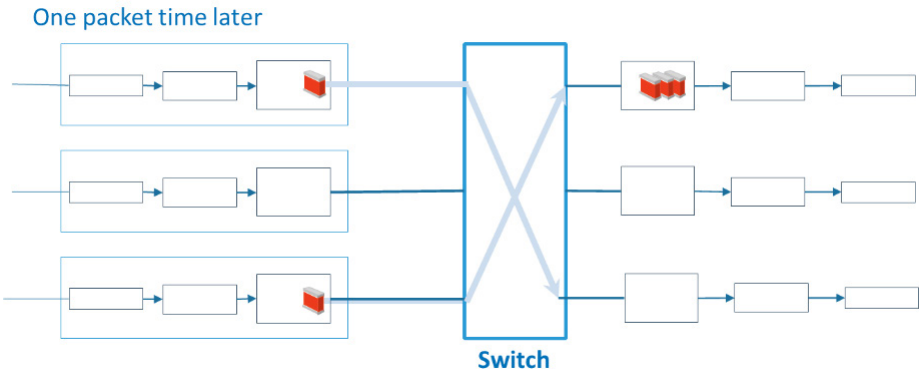


Figura 5. Creación de congestión

Fuente: elaboración propia. Modificado de Kurose (2009)

En cualquiera de los casos en que la velocidad de entrada de datos o el procesamiento de los mismos en el router sea menor a la transferencia de ellos, se generara encolamiento de paquetes. Sin embargo, las velocidades de procesamientos de estos equipos, está pensada y diseñada para que sea mucho mayor.

2. IPv4

Internet Protocol versión 4, está definido en el RFC-791. Es uno de los protocolos más importantes que se usa en la capa 3 de red y de Internet, su primera versión apareció en 1983 con Arpanet. Las direcciones IP están compuestas por una longitud de 32 bits separados 4 veces en grupos de 8 cada uno, lo que nos da un total de 2^{32} direcciones posibles, mal contadas son más de 4000 millones. Las direcciones IP se denotan de la siguiente manera 192.168.0.0, cada grupo de 8 bits separado por puntos donde pueden ir de 0 a 255 en decimal.

Siendo su expresión en binario de la siguiente manera:

11000000.10101000.00000000.00000000

Cada una de estas direcciones debe ser única e irrepetible para cada dispositivo en el mundo a los ojos de Internet, sin embargo, dado el número tan grande de dispositivos que en este momento tiene el mundo, se han creado algunas técnicas para tratar de utilizar direcciones de forma repetitiva a través de NAT, que lo explicaremos más adelante.

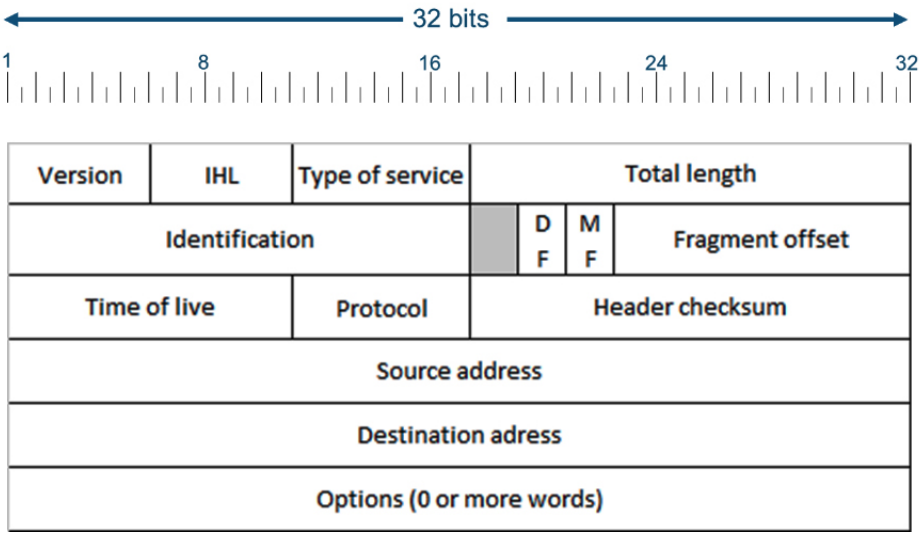


Figura 6. Datagrama IPv4

Fuente: elaboración propia. Modificado de Kurose (2009)

En la capa 3 de red nos referimos a los paquetes como datagramas, la figura 6 nos enseña el datagrama IPv4.

- **Versión:** Los 4 bits que hay aquí nos dicen la versión del datagrama IP.
- **Longitud de la cabecera:** Los 4 siguientes bits nos indican donde comienza nuestro datagrama, pues algunas de las variables que vemos aquí, puede cambiar.
- **Tipo de servicio:** El tipo de servicio va enfocado a una prioridad de los paquetes, no es lo mismo enviar datagramas de internet, que datagramas de voz IP, los cuales tienen que llegar más rápido y sin pérdidas.
- **Longitud del datagrama:** La longitud del datagrama es el encabezado más los datos, este tiene 16 bits los que nos da un tamaño de 65535 bytes sin embargo los datagramas no superan los 1500 bytes, es muy extraño un paquete más grande.
- **Identificadores:** Este campo se refiere a la fragmentación de los datagramas, en la versión 6 no se permite esta fragmentación, después lo veremos.
- **Tiempo de Vida:** Este se incluye con el fin de que los datagramas no se queden dando vueltas en la red, en caso de un loop y puedan ser descartados al final cuando su valor llegue a cero.

- **Protocolo:** Este valor solo se asigna cuando el datagrama es entregado, por ejemplo si allí hay un valor de 6 quiere decir que los datos pasan a TCP y cuando hay un valor de 17 a UDP (Puede descargar la lista completa de IANA Protocol Numbers 2009).
- **Comprobación de cabecera:** Este valor es introducido en cada router y se refiere a una comprobación para detectar errores de bit en un datagrama.
- **Direcciones de origen y destino:** Cuando el router crea el datagrama, el host tiene que indicar su destino y de paso coloca su dirección para que pueda ser respondido
- **Opciones:** En un principio se pensó en que el datagrama pudiera ser variable, sin embargo con el tiempo se dieron cuenta que estos cambios pueden hacer que cambie de manera dramática el tiempo de procesamiento del mismo, en la versión 6 este campo fue cancelado.
- **Datos:** Este campo es el más importante, pues aquí van los bytes a ser entregados.

2.1. Fragmentación del datagrama IP

Los datagramas se tienen que fragmentar para poder entregarlos a la capa 2, hay algunos protocolos que no se pueden transportar. Por ejemplo, las tramas Ethernet pueden transportar hasta 1500 bytes de datos y hay otros donde ese tamaño es menor. Esta medida se conoce como MTU (máximum transmission unit), sin embargo, esto no es problema y por eso se creó el campo de Opciones en la cabecera, para poder hacer esto y tener referencia de ello, y de paso los diseñadores del datagrama pasaron parte del inconveniente a la siguiente capa, la de transporte quien es la que tiene que reunir los fragmentos del datagrama IP nuevamente.

La figura 6 nos muestra un claro ejemplo con un datagrama de tamaño de 4.000Bytes, el cual tiene que ser fragmentado para ser enviado por la red. Lo primero que se tiene que hacer es restar los 20bytes de la cabecera y nos quedan 3.980bytes del datagrama original y este lo tenemos que dividir en tres datagramas más pequeños de 1500Bytes, abajo veremos la tabla que se hizo para mostrar la división del datagrama.

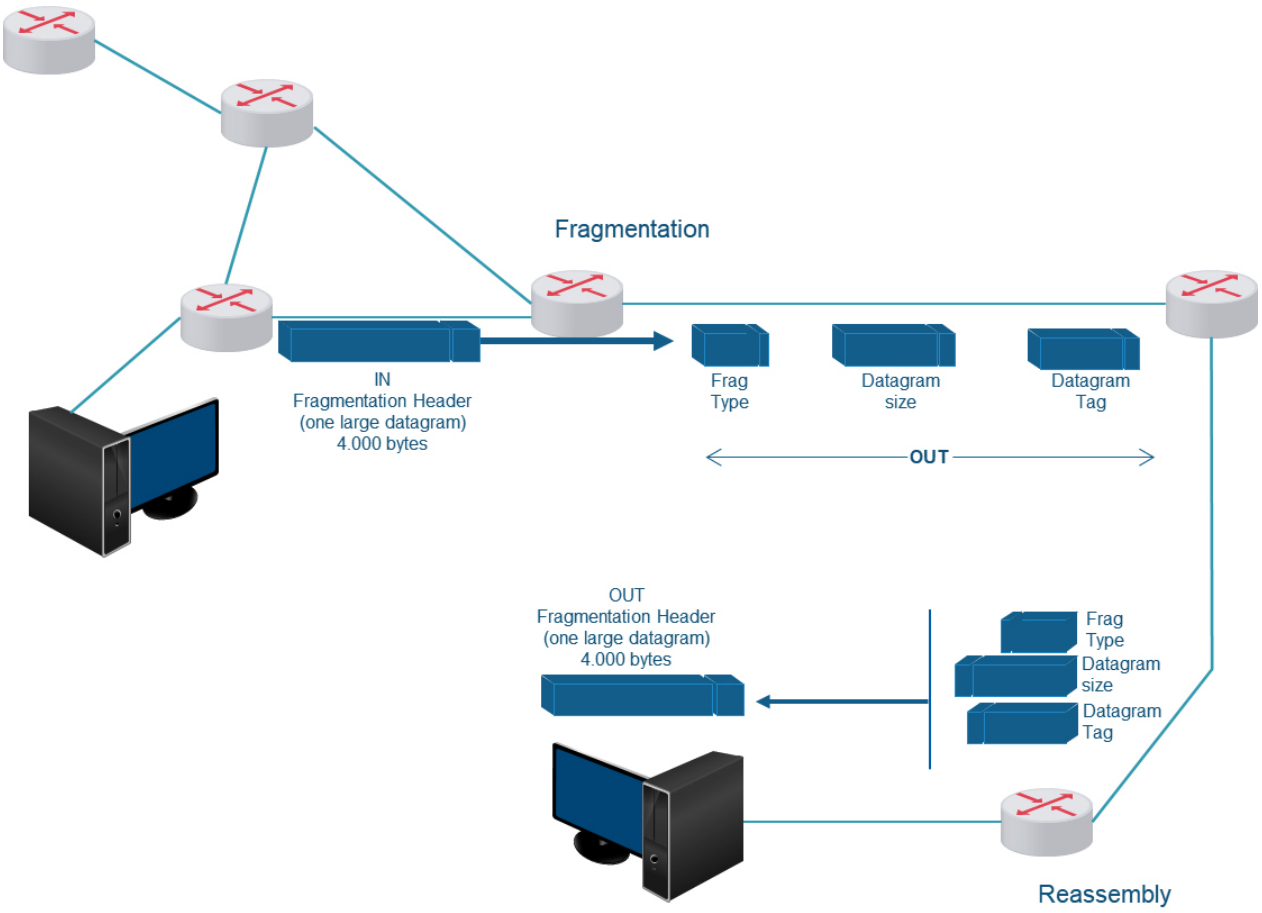


Figura 7. Fragmentación de un datagrama

Fuente: elaboración propia. Modificado de Kurose (2009)

3. IPv6

IPv6 es la extensión de IPv4, o como dicen los expertos la versión mejorada, sin embargo, en sus inicios no se busco hacer la mejora en cuanto a flexibilidad o seguridad, el problema se radico en el numero de host que podían contener las direcciones IPv4 que solo llegaban a mas de 4000 millones y de 1990 a 1998 vieron una gran expansión y uso de ellas con el internet. Los estudios de la IETF (Internet Engineering Task Force), hacían cuenta que en los 2008 al 2018 las direcciones IPv4 se acabarían. Para más información acerca del procotolo IPv6, pueden referirse al RFC-2460, que explica en detalle su uso y funcionamiento.

El cambio más grande y significativo esta en pasar de direcciones de 32 a 128bits, logrando un número considerable de host existentes para conectar la gran cantidad de dispositivos que se tienen hoy en el mundo, se dice que en un futuro hasta las neveras tendrán su propia IP.

IPv4 entregaba 4.294.967.296 (232) direcciones de host diferentes, un número grande en su época cuando inició y la Internet no estaba masificada en el mundo. En cambio, IPv6 permite 340.282.366.920.938.463.463.374.607.431.768.211.456 (2128 ó 340 sextillones de direcciones), lo cual nos permitirá a futuro entregarle hasta 1.000.000.000 IPs por persona.

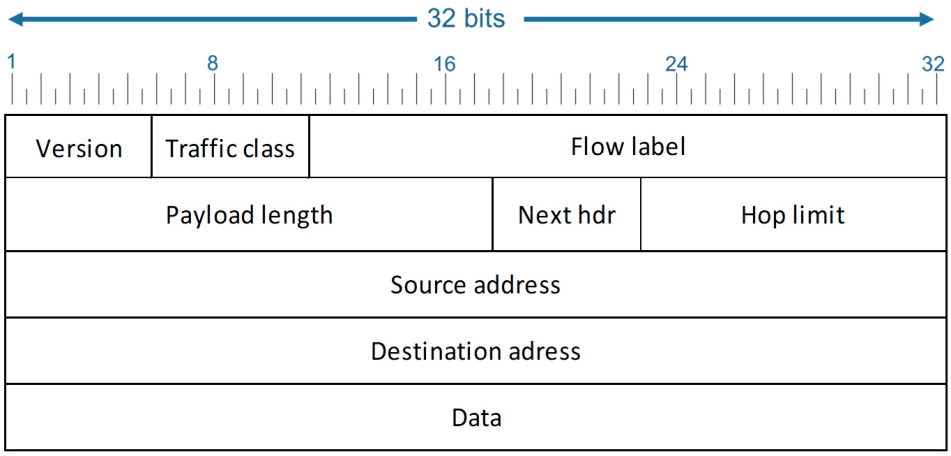


Figura 8. Datagrama IPv6

Fuente: elaboración propia. Modificado de Kurose (2009)

La figura que vemos nos muestra el datagrama de IPv6, muy parecido al de IPv4, claro está con unas pequeñas variantes.

- **Versión:** Los 4 bits que hay aquí nos dicen la versión del datagrama IP, en este caso el 6.
- **Clase de tráfico:** Muy parecido al tipo de servicio, antes encontrado en esta casilla.
- **Etiqueta de flujo:** Este campo de 20bits, se usa para identificar el flujo de datagramas.
- **Longitud de carga:** Este valor entrega el número de bytes del datagrama.
- **Siguiente cabecera:** Esta casilla nos entrega el dato, a cual protocolo se entregara el contenido del datagrama, ejemplo TCP o UDP.
- **Limite de saltos:** Funciona igual que el TTL, del IPv4.

- **Direcciones origen y destino:** Son las direcciones IP que envían y reciben al datagrama.
- **Datos:** Es la parte útil del datagrama cuando la capa de red lo entregue a la siguiente capa, esta tomara sus datos y los usara.

Ipv6 no tiene fragmentación, la fragmentación y el ensamble son procesos que quitan tiempo, por ende se retiraron de la cabecera, y ayuda acelerar el proceso de envío y reenvío de datagramas.

La suma de comprobación de la cabecera también fue retirada por motivos similares y aparte del retardo, las otras capas que intervienen, tanto enlace como transporte también realizan comprobaciones de estado de los datos y paquetes, entonces se pensó que era redundante el control de flujo.

La migración de este par de protocolos aun se realiza en el mundo, y estos trabajan de forma conjunta, pues ni se ha retirado IPv4 ni se ha impuesto a la fuerza IPv6, para que estos 2 protocolos vivan en conjunto se han creado varios sistemas donde comparten información e interoperan en las redes. Se tiene el método de la pila dual, donde routers que trabajan con IPv4, transporten datagramas por tuneles de IPv6.

4. VLSM (Variable Length Subnet Mask), o Máscaras de subred de tamaño variable

Esta es una técnica usada para aprovechar el número de IP definidas en IPv4. La idea de este método es utilizar la máxima cantidad de IP disponible dentro un rango que se tenía en redes privadas, sin desperdiciarlas o no dándoles uso para decirlo de otra manera. Después de esto nació también NAT, que significa (Network Address Translation) y traduce translación de direcciones de red, lo cual simplifico un poco el problema a la falta de IPs.

Para comenzar VLSM, tenemos que recordar que las direcciones IP no vienen solas, estas vienen acompañadas de una máscara de red la cual determina si una dirección es válida o no.

Veamos un par de ejemplos y de paso conocemos las IP privadas.

CLASE A:

(10.0.0. 0 a 10.255.255.255) RED 8bits, Mascara: 255.0.0.0

Ejemplo Red: 10.0.0.0 255.0.0.0

Ejemplo Host: 10.0.0.1 255.0.0.0

CLASE B:

(172.16.0. 0 a 172.31.255.255) RED 16bits, Mascara: 255.255.0.0

Ejemplo Red: 172.17.0.0 255.255.0.0

Ejemplo Host: 172.17.0.1 255.255.0.0

CLASE C:

(192.168.0. 0 a 192.168.255.255) RED 24bits, Mascara: 255.255.255.0

Ejemplo Red: 192.168.18.0 255.255.255.0

Ejemplo Host: 192.168.18.1 255.255.255.0

Y como sabemos ¿cuáles son las redes y cuales los host?

Para esto se tiene que hacer una operación booleana entre la dirección IP y la máscara de red con los últimos bits en binario, hagamos el ejemplo con esta última.

<table>
<tr><td>Ejemplo Red</td><td>192.168.18.0</td><td colspan="2">255.255.255.0</td></tr>
<tr><td rowspan="2">Binario</td><td rowspan="2">192.168.18.00000000</td><td>RED</td><td>HOST</td></tr>
<tr><td colspan="2">255.255.255.00000000</td></tr>
</table>

Para determinar si es un host valido, se hace una AND entre los últimos dígitos en red en este caso, los últimos entre IP y mascara.

Ejemplo Host:

192.168.18. 00000001 255.255.255.00000000
255.255.255.00000000
XXX.XXX.XXX.00000001 Esta es una IP válida.

Lo podemos organizar de esta manera.

- Si termina en 1 binario y está acompañado de otros 0, es un IP válida.
- Si todos sus dígitos son 1, es una IP broadcast.
- Si todos sus dígitos son 0, es una IP de red.

Veámoslo!!!

Ejemplo Red:

192.168.18. 00000000 255.255.255.00000000
255.255.255.00000000
XXX.XXX.XXX.00000000 Esta es una IP red.

Ejemplo Host:

192.168.18. 255
192.168.18. 11111111 255.255.255.00000000
255.255.255.00000000
XXX.XXX.XXX.11111111 Esta es una IP de broadcast.

Con esta mascara de 24bits, tenemos 8 bits para host. Estos los calculamos de la siguiente manera.

HOST VALIDOS (2^n - 2) ((2^8 = 256(El cero se cuenta)) – 2) = 254

Referencias

Kurose, J. (2009). *Computer Networking: A Top Down Approach Featuring the Internet 5rd edition*. Addison - Wesley, 2009.

Boronat, S. F. y Montagud, C. M. (2013). *Direccionamiento e interconexión de redes basada en TCP/IP: IPv4/IPv6, DHCP, NAT, Encaminamiento RIP y OSPF*. Valencia, ES: Editorial de la Universidad Politécnica de Valencia. Recuperado de: http://www.ebrary.com.loginbiblio.poligran.edu.co:2048

INFORMACIÓN TÉCNICA

FACULTAD DE
INGENIERÍA, DISEÑO
E INNOVACIÓN

Módulo: Telecomunicaciones

Unidad 4: Funcionamiento de la capa de red

Escenario 7: La capa de red

Autor: John Alirio Olarte Ramos

Asesor Pedagógico: Juan Felipe Marciales

Diseñador Gráfico: Karim Gaitán

Asistente: María Avilán