

Ampliación de Sistemas Operativos y Redes

Resumen de la asignatura.

Curso 2018/19

Índice general

1. Redes	5
1.1. Introducción, OSI vs TCP/IP	5
1.2. Protocolo IPv4	6
1.2.1. Formato del mensaje IP	6
1.2.2. Direccionamiento	7
1.3. Protocolo ARP	9
1.4. Protocolo ICMP	10
1.4.1. ECHO Request/Reply	10
1.5. Protocolo DHCP	12
1.5.1. Mensajes DHCP	13
1.5.2. Datagrama DHCP	13
1.6. Protocolo TCP	14
1.6.1. Datagrama TCP	15
1.6.2. Fases de conexión	16
1.6.3. Ventana deslizante	19
1.6.4. Control de errores	21
1.6.5. Temporizadores TCP	24
1.6.6. Control de flujo	25
1.6.7. Control de la congestión	26
1.7. Servicios de Red: Filtrado de paquetes.	29
1.7.1. Firewall y filtrado de paquetes.	29
1.7.2. Iptables.	29
1.8. DNS	33
1.8.1. Zonas y dominios.	33
1.8.2. Datagrama DNS	34
1.8.3. Características del protocolo DNS	34
1.9. Protocolo IPv6	37
1.9.1. IPv4 vs IPv6	37
1.9.2. Direcciones IPv6	38
1.9.3. Datagrama IPv6	40
1.10. Protocolo ICMPv6	42
1.10.1. Mensajes ICMPv6	42
1.11. Encaminamiento en Internet	44
1.11.1. Técnicas de Encaminamiento	44
1.12. Vector de distancias - Protocolo RIP	45
1.12.1. Funcionamiento	45
1.12.2. Problemas	46
1.12.3. Protocolo RIP	47

1.13. Estado de los enlaces - Protocolo OSPF	49
1.13.1. Protocolo OSPF	49
1.14. Vector de Rutas - Protocolo BGP	51
1.14.1. Protocolo BGP	51

Capítulo 1

Redes

1.1. Introducción, OSI vs TCP/IP

Durante la asignatura de redes estudiamos el modelo OSI, sin embargo durante ésta asignatura nos centraremos en el estudio del modelo TCP/IP.

El modelo de Internet (TCP/IP) fue desarrollado en 1972 por el Departamento de Defensa de los Estados Unidos como una solución a un problema práctico de ingeniería.

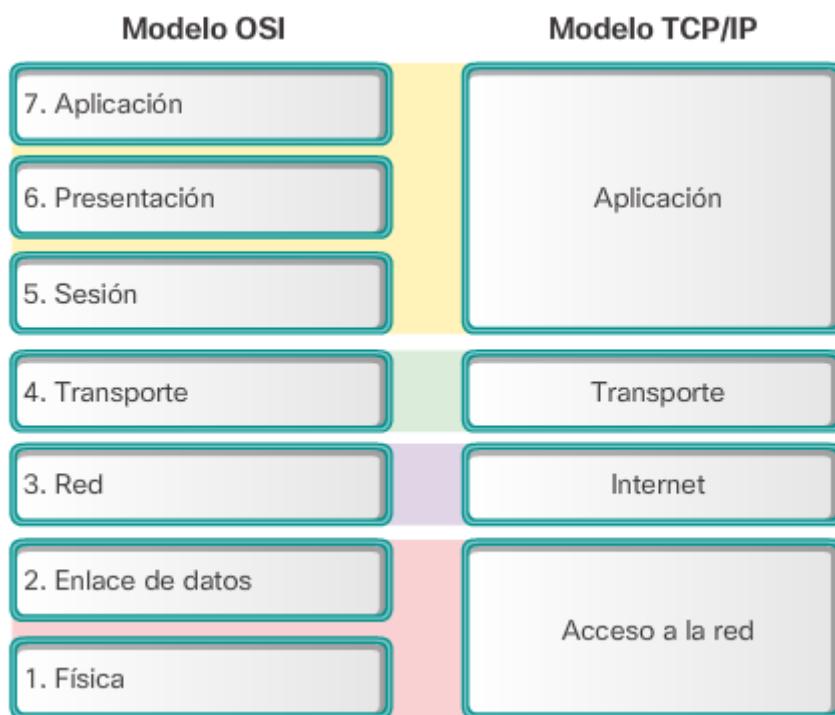
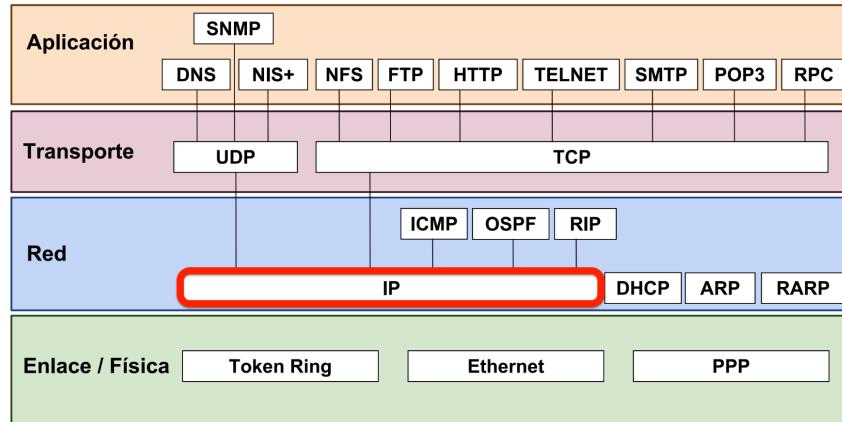


Figura 1.1: Comparativa de las capas de los modelos

En cambio, el modelo OSI(Open System Interconnection) fue propuesto como una solución teórica a los problemas de incompatibilidad entre redes.

Hoy en día es el modelo TCP/IP el que realmente se usa.

1.2. Protocolo IPv4



IP (Internet Protocol) es un protocolo de comunicación de datos digitales que transfiere paquetes a través de distintas redes físicas. Sus funciones básicas son:

- Encaminamiento: Saber por dónde tiene que enviar un mensaje.
- Direccionamiento: Cómo nombrar las máquinas que están dentro de la red.
- Fragmentación y Reensamblado: Traducir las tramas entre tecnologías distintas.

El protocolo IP es no orientado a conexión y no fiable, ésto quiere decir:

- No detecta paquetes erróneos.
- No recupera paquetes perdidos.
- No garantiza que los paquetes lleguen en orden.
- No garantiza la detección de paquetes duplicados.

1.2.1. Formato del mensaje IP

El datagrama IP está formado por una cabecera IP seguida de un campo de datos. La cabecera para IPv4 es de la siguiente manera.

El campo opciones puede indicar:

- Encaminamiento desde el origen: lista de los encaminadores hasta llegar al destino. Usado para depuración o para saturar redes.
- Registro de ruta: Cuando el datagrama llega al destinatario, tiene una lista de todos los routers por los que ha pasado.
- Salto de tiempo: Cada router pone una marca de tiempo, se usa para medir el rendimiento de la red.

Formato de la Cabecera IP (Versión 4)								
0-3	4-7	8-15	16-18	19-31				
Versión	Tamaño Cabecera	Tipo de Servicio		Longitud Total				
Identificador		Flags		Posición de Fragmento				
Tiempo de Vida	Protocolo		Suma de Control de Cabecera					
Dirección IP de Origen								
Dirección IP de Destino								
Opciones			Relleno					

1.2.2. Direccionamiento

Una dirección IP es un número que identifica a una Interfaz en red de un dispositivo que utilice el protocolo IP. La dirección IP no debe confundirse con la dirección MAC, que es un identificador de 48 bits para identificar de forma única la tarjeta de red y no depende del protocolo de conexión utilizando la red.

Las direcciones IPv4 se expresan mediante un número binario de 32 bits y se suelen expresar como números de notación decimal: se dividen los 32 bits de la dirección en cuatro octetos. El valor decimal de cada octeto está comprendido en el intervalo de 0 a 255. Ejemplo: [10.128.1.253]

Tipos de direcciones IP.

Las direcciones se dividen en dos campos de longitud variable: NetID (que identifica la red) y HostID (que identifica una máquina dentro de la red). Éstos rangos se indican mediante la máscara de red, que es un número que representa el número de bits de la NetID.

- Unicast: es el concepto más común en IP, se utiliza para un sólo dispositivo. [ej. 147.96.2.4]
- Multicast: se utiliza para un grupo de receptores interesados. Se pueden utilizar direcciones comprendidas entre 224.0.0.0 - 239.255.255.255. [ej. 239.5.27.2]
- Broadcast
 - Limitada: usada para enviar a todas las máquinas de mi red LAN. [ej. 255.255.255.255]
 - Dirigida: usada para enviar a todas las máquinas de una determinada red. [ej. 147.96.255.255]
- Anycast: envía el datagrama a la máquina más cercana de un grupo.

La dirección IP debe ser única. Es la organización IANA quien se encarga de asignar las direcciones IP. Inicialmente se organizaron las direcciones IP en clases.

Clase	Bits iniciales	Intervalo (*)
A	0	0.0.0.0 (**)-127.255.255.255
B	10	128.0.0.0 - 191.255.255.255
C	110	192.0.0.0 - 223.255.255.255
D (Multicast)	1110	224.0.0.0 - 239.255.255.255
E (experimental)	1111	240.0.0.0 - 255.255.255.255

El principal problema de esta forma de organización es el desperdicio de direcciones, por ello actualmente se utiliza una organización sin clases (CIDR).

CIDR

Classless Interdomain Routing, permite un uso más eficiente de las direcciones IP mediante la utilización de bloques de tamaños arbitrarios.

La notación CIDR permite expresar fácilmente las direcciones IP:

Consta de una dirección IPv4 de 32 bits seguida de una barra y un número, que indica el número de bits que identifican al bloque.

122.233.2.1/24

Ésta dirección indica que los primeros 24 bits de la dirección son los que identifican al bloque (Prefijo de red), y los otros 8 menos significativos son los que identifican al host.

Direcciones reservadas

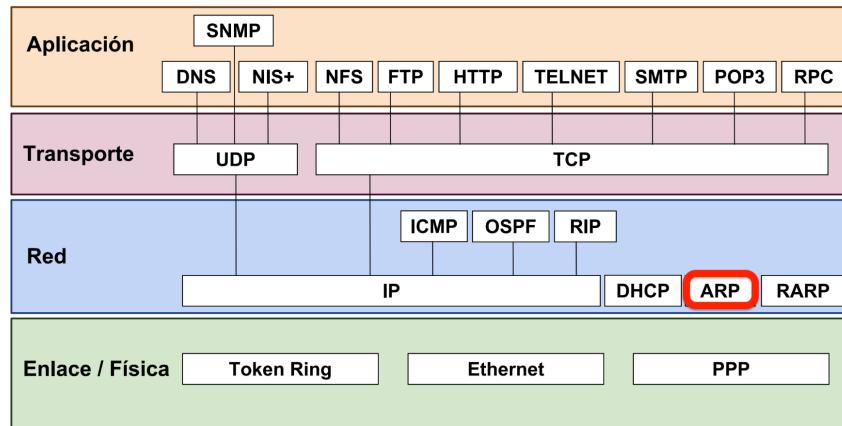
El **identificador de host** tiene dos excepciones, según el valor de sus bits:

1. **Todo a ceros:** indica que se trata de la dirección que identifica a la red (no de ninguna máquina), por tanto nunca se puede utilizar como dirección de destino.
2. **Todo a unos:** indica que se trata de una dirección de broadcast, y al usarse como dirección de destino enviará el paquete a todas las máquinas de la red local.

A parte, existen otras direcciones reservadas:

- **10.x.x.x** red privada de clase A.
- **172.16.0.0 - 172.31.255.255** son 16 redes privadas de clase B.
- **192.168.x.x** son 256 redes privadas de clase C
- **127.x.x.x** Direcciones de Loopback (reenvían los paquetes a la misma máquina)
- **224.0.0.xxx** son direcciones multicast reservadas, por ejemplo:
 - **224.0.0.1** Todos los hosts.
 - **224.0.0.2** Todos los routers.

1.3. Protocolo ARP



Un datagrama IP tiene la siguiente forma.



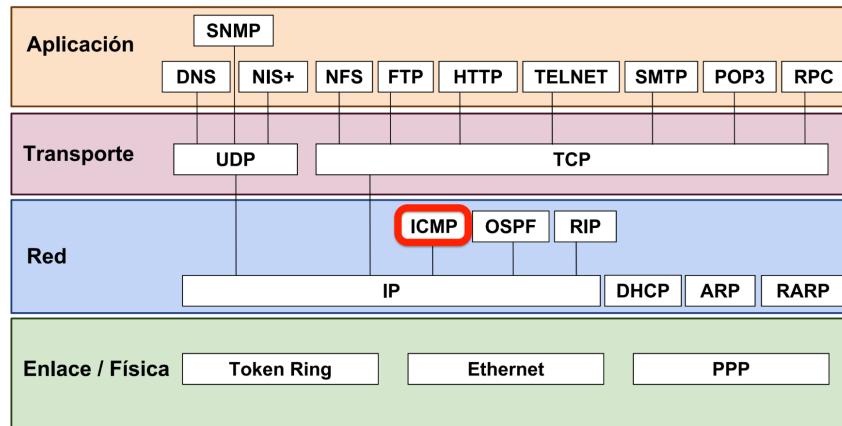
A la hora de ser enviado, se encapsula **dentro del campo de datos** de otra trama más grande, la trama ARP.



ARP se encarga de encontrar la dirección MAC que corresponde a una determinada dirección IP. Para ello se envía un paquete (ARP request) a la dirección de difusión de la red (broadcast, MAC = FF FF FF FF FF FF) que contiene la dirección IP por la que se pregunta, y se espera a que esa máquina responda (ARP reply) con su dirección MAC.

Cada máquina mantiene una tabla ARP en la que guarda las direcciones IP de las últimas máquinas con las que ha conectado, junto con su dirección MAC.

1.4. Protocolo ICMP



ICMP (Internet Control Message Protocol) es un protocolo de mensajes que viajan dentro del datagrama IP. Se usa para enviar mensajes de error, indicando por ejemplo que un router o host no puede ser localizado. También puede ser utilizado para transmitir mensajes de información.

Algunos tipos de mensajes ICMP:

- De información
 - Echo Reply
 - Echo Request
 - Redirect
 - Router Solicitation
 - Router Advertisement
- De error
 - Destination Unreachable
 - Source Quench
 - Time Exceeded
 - Parameter Problem

1.4.1. ECHO Request/Reply

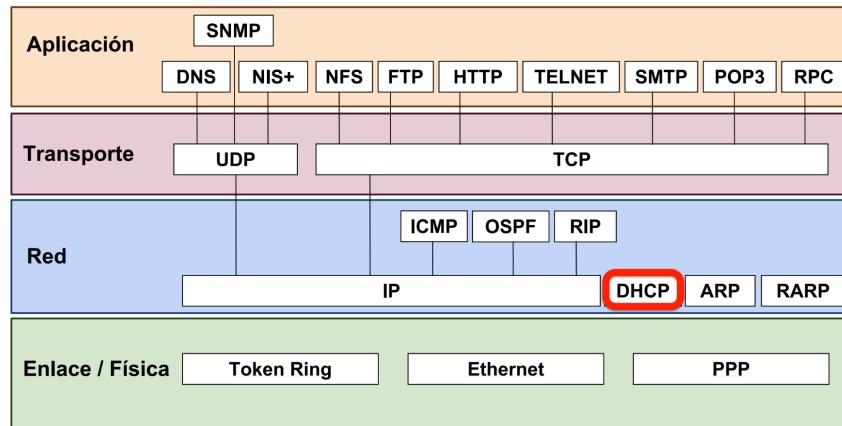
Como mensajes a destacar tenemos ECHO Request y ECHO Reply, que son los que utiliza la orden "ping" para comprobar la conexión.

Éstos mensajes tienen el siguiente formato:

Tipo = 0	Código = 0	Checksum
Identificador		Número de secuencia
Datos :::		

- **Identificador:** Permite establecer correspondencia entre solicitud (Request) y respuesta (Reply); ambos con el mismo identificador.
- **Secuencia:** También se utiliza para establecer la correspondencia entre solicitud y respuesta, cuando se envían varios Echo Requests consecutivos con el mismo identificador.
- **Datos:** Un número determinado de bytes aleatorios.

1.5. Protocolo DHCP



DHCP (Dynamic Host Configuration Protocol) es un protocolo que permite la configuración dinámica de direcciones IP y máscaras de red, router predeterminado, servidor DNS, y otros parámetros de configuración de red.

Este servidor posee una lista de direcciones IP dinámicas y las va asignando a los clientes conforme éstas van quedando libres, sabiendo en todo momento quién ha estado en posesión de esa IP, cuánto tiempo la ha tenido y a quién se la ha asignado después. Así los clientes de una red IP pueden conseguir sus parámetros de configuración automáticamente.

Algunas de sus **características** son:

- Es un protocolo de red de tipo **cliente/servidor** sobre UDP en los puertos 67 (servidor) y 68 (cliente).
- Tiene un mecanismo de **control de errores** basado en sumas de comprobación, temporizadores y retransmisiones.
- Puede proveer de una servidor **TFTP** al dispositivo cliente. TFTP se utiliza para transferir pequeños archivos entre ordenadores en una red, como imágenes de arranque.
- En redes grandes con muchos enlaces, un servidor DHCP es ayudado por **DHCP relay agents** situados en routers intermedios. Estos agentes retransmiten los mensajes entre cliente y servidor situados en distintas subredes.

1.5.1. Mensajes DHCP

Estos son algunas de las operaciones que permite el protocolo.

- **DHCPDISCOVER:** Mensaje del cliente (broadcast) para descubrir los servidores disponibles (puede contener la última dirección IP asignada).
- **DHCPOFFER:** Respuesta de los servidores, con una oferta de parámetros de configuración. Puede recibirse más de una.
- **DCHPREQUEST:** Petición de oferta del cliente (broadcast, para notificar a todos los servidores) o extensión del tiempo de cesión. El servidor seleccionado se especifica en una opción (Server Identifier, código 54).
- **DHCPACK:** Mensaje de confirmación (broadcast) y cierre desde el servidor hacia el cliente indicando los parámetros definitivos.
- **DCHPRELEASE:** Mensaje del cliente para informar al servidor de que ha finalizado el uso de la dirección IP.

1.5.2. Datagrama DHCP

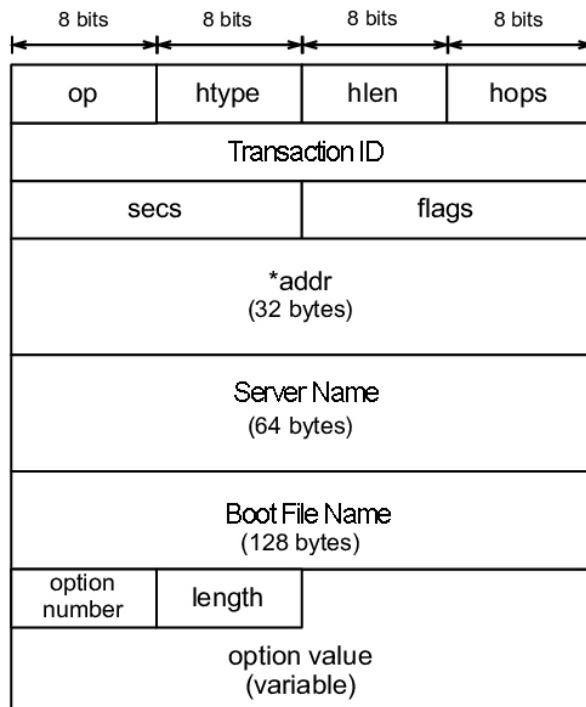
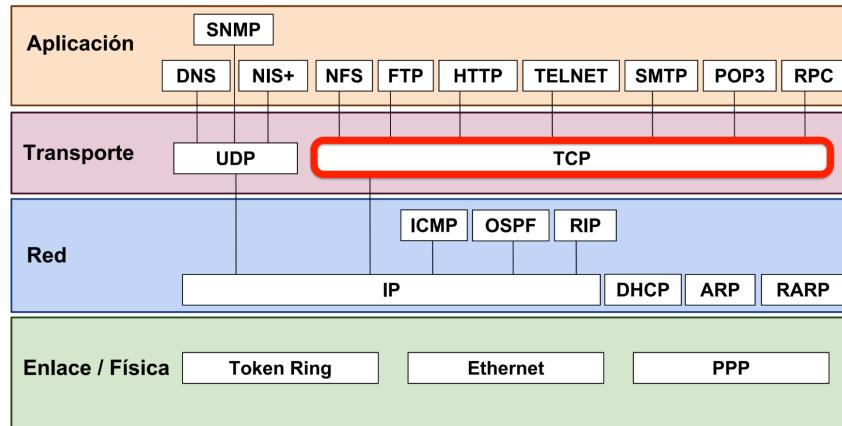


Figura 1.2: *addr: se guardan algunas direcciones como las de cliente y servidor.

1.6. Protocolo TCP



TCP (Transmission Control Protocol) es un protocolo de transporte fiable y orientado a conexión (a diferencia de UDP, que es no fiable y no orientado a conexión). También proporciona un mecanismo para distinguir distintas aplicaciones dentro de una misma máquina, a través del concepto de puerto.

Características de TCP

- Reordena los segmentos procedentes del protocolo IP y permite comenzar y finalizar la comunicación de forma consensuada entre ambas máquinas. **Orientado a conexión.**
- Monitorea el flujo de datos y así evita la saturación de la red. **Ventana Deslizante.**
- Permite que los datos se formen en segmentos de longitud variada para entregarlos al protocolo IP.
- Sirve para establecer la comunicación entre procesos de distintas máquinas. **Puertos.**
- Permite **multiplexar** los datos, es decir, que la información que viene de diferentes fuentes pueda circular simultáneamente por el mismo medio.

1.6.1. Datagrama TCP

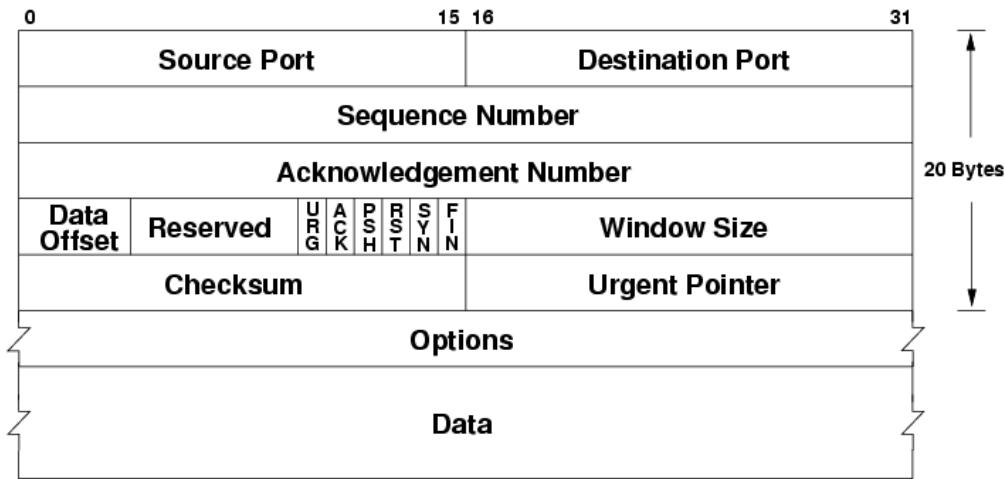


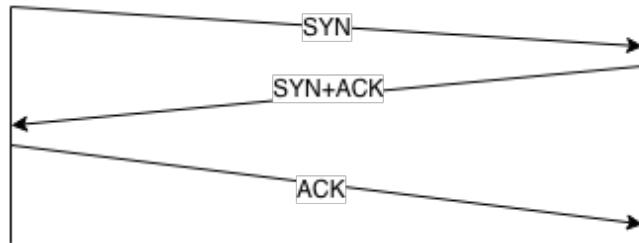
Figura 1.3: Se divide en dos partes: datos (Data) y cabecera (Resto de campos).

- **Source/Destination Port**: Identifican ambos extremos de la conexión.
- **Sequence/Acknowledgement Number**: Números de secuencia y confirmación expresados en bytes.
- **Reserved**: No se usa y se pone a 0.
- **Data offset**: Longitud de la cabecera.
- **Flags**
 - URG: El segmento transporta datos urgentes (URG=1) desde el primer byte hasta el nº de byte especificado en el campo puntero urgente. TCP notifica a la aplicación de los datos urgentes (mediante la señal SIGURG), y ésta se encarga de tratarlos.
 - ACK: El segmento contiene un número de confirmación válido (ACK=1). Todos los segmentos de una conexión TCP, excepto el primero, llevan ACK=1.
 - PSH: Los datos deben ser entregados inmediatamente a la aplicación (PSH=1), o pueden almacenarse en el buffer (PSH=0).
 - RST: Utilizado para abortar una conexión.
 - SYN: Utilizado en el establecimiento de la conexión y sincronizar los números de secuencia iniciales.
 - FIN: Utilizado en la finalización de la conexión.
- **Window Size**: Tamaño de la ventana.
- **Checksum**: Suma de comprobación.
- **Urgent Pointer**: Puntero que identifica el último byte de datos urgentes.
- **Options**: Campo de opciones de longitud variable.
- **Data**: Campo de datos de longitud variable.

1.6.2. Fases de conexión

Ya sabemos que TCP es un protocolo orientado a conexión, ahora vamos a ver detenidamente como se comunican dos máquinas.

Fase de Establecimiento



Una máquina (**servidor**) abre socket en un determinado puerto TCP y se queda a la escucha de nuevas conexiones (apertura pasiva).

Otra máquina (**cliente**) realiza una apertura activa de un puerto enviando un paquete SYN al servidor como parte de la negociación en tres pasos. En el lado del servidor se comprueba si el puerto está abierto, es decir, si existe algún proceso escuchando en ese puerto.

En caso de no estarlo, se envía al cliente un paquete de respuesta con el bit RST activado, lo que significa el rechazo del intento de conexión.

En caso de que sí se encuentre abierto el puerto, el lado servidor respondería a la petición SYN válida con un paquete SYN/ACK. Finalmente, el cliente debería responderle al servidor con un ACK, completando así la negociación en tres pasos (SYN, SYN/ACK y ACK) y la fase de establecimiento de conexión.

Es interesante notar que existe un número de secuencia generado por cada lado, ayudando de este modo a que no se puedan establecer conexiones falseadas (spoofing).

SYN Flood

Es una vulnerabilidad en el protocolo que consiste en enviar una gran cantidad de segmentos TCP con el flag SYN activado, saturando el servidor (ataque DoS), ya que asigna recursos a cada intento de conexión. Para evitarlos se puede:

- Limitar el número de conexiones.
- Aceptar conexiones sólo de IP's confiables.
- Retrasar la asignación de recursos usando SYN cookies.

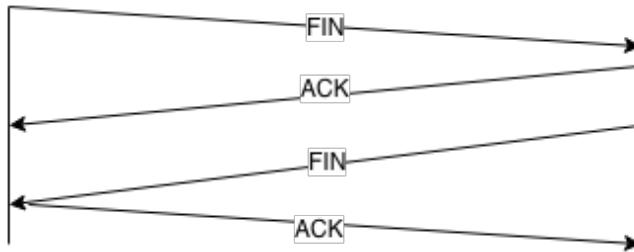
Fase de Transferencia

Durante la etapa de transferencia de datos, una serie de mecanismos claves determinan la fiabilidad y robustez del protocolo. Entre ellos están incluidos el uso del **número de secuencia** para ordenar los segmentos TCP recibidos y detectar paquetes duplicados, **checksums** para detectar errores, **temporizadores** para detectar pérdidas o retrasos y **ventanas deslizantes** para el control de flujo de datos.

Fase de Finalización

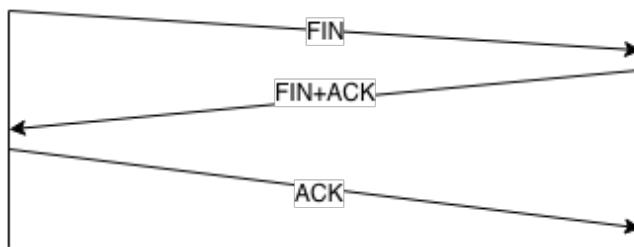
La fase de finalización puede realizarse de dos maneras:

- **Finalización de 4 vías:**



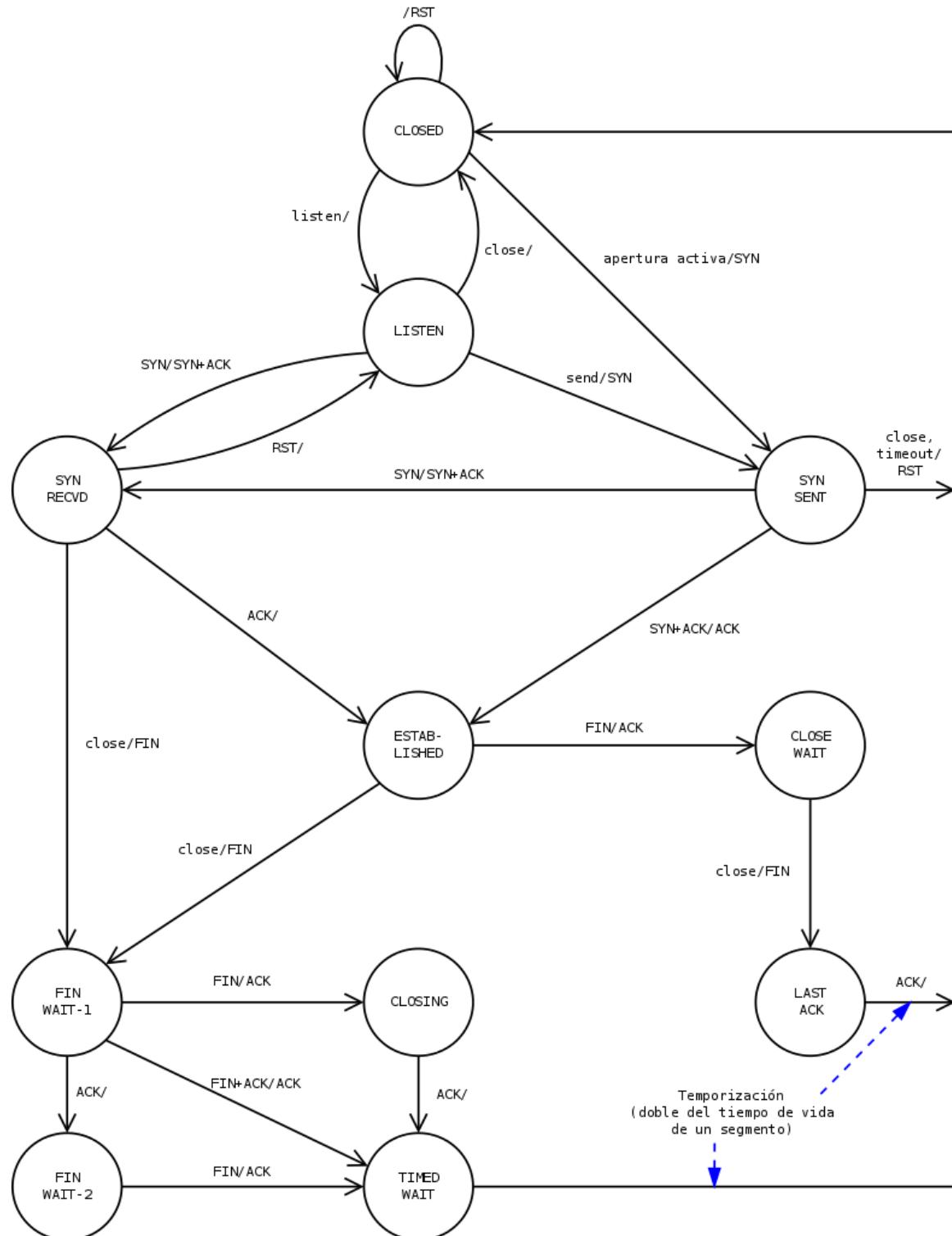
1. El cliente deja de enviar datos al servidor, y le envía un paquete con el flag FIN activado.
2. El servidor le responde con un paquete con el flag ACK.
3. Cuando el servidor deja de tener datos para enviar, envía otro paquete con el flag FIN.
4. El cliente le responde con ACK, tras lo cual se cierra la conexión.

- **Finalización de 3 vías:**



1. El cliente deja de enviar datos al servidor, y le envía un paquete con el flag FIN activado.
2. El servidor le responde con un paquete con los flags FIN y ACK activados y deja de enviar datos.
3. El cliente le responde con ACK, tras lo cual se cierra la conexión.

Diagrama de estados



1.6.3. Ventana deslizante

TCP consigue transferencia fiable mediante el uso de la **ventana deslizante**.

Parada y Espera vs. Ventana Deslizante

En los protocolos de **parada y espera**, cuando el emisor envía un segmento, tiene que esperar a recibir un ACK antes de poder enviar el siguiente segmento.

Si después del timeout no llega el ACK, el emisor vuelve a enviar el segmento. Esto implica que un transmisor solo puede tener un elemento pendiente de reconocimiento, lo cual es bastante ineficiente.

Los protocolos de **ventana deslizante** por el contrario, pueden enviar varios segmentos sin esperar a recibir las confirmaciones de una en una.

La ventana deslizante permite:

- La recepción de segmentos duplicados.
- La retransmisión de segmentos erróneos o perdidos.
- La recepción de segmentos fuera de orden.



Figura 1.4: Intercambio de paquetes mediante ventana deslizante.

Emisor

- Almacena los elementos pendientes de reconocimiento [marcados en rojo en el ejemplo]. El límite inferior de la ventana indica el primer segmento sin confirmar (si lo hay), o el siguiente segmento a enviar (si todos los segmentos están confirmados).
- Almacena también los elementos que pueden ser enviados [marcados en amarillo].
- El tamaño de la ventana siempre es igual al tamaño máximo [7 en el ejemplo]. Excepto cuando la aplicación no tiene tantos datos para enviar, en cuyo caso es menor.

Receptor

- Almacena los elementos pendientes de ser consumidos por la aplicación [marcados en rojo en el ejemplo]. El límite inferior de la ventana indica el primer segmento sin ser consumido (si lo hay), o el siguiente segmento a recibir (si todos los segmentos están consumidos).
- Almacena también los elementos que pueden ser recibidos [marcados en amarillo].
- El tamaño de la ventana siempre es igual al tamaño máximo [7 en el ejemplo]. Excepto cuando la aplicación no tiene tantos datos para recibir, en cuyo caso es menor.
- Una vez los segmentos son recibidos, el receptor envía un ACK, cuyo número indica el siguiente segmento a recibir.
- Para enviar los ACK se utiliza la técnica de **Piggybacking**, que consiste en incluir el ACK en otro segmento de datos que el receptor quiera enviar al emisor (ya que la comunicación es bidireccional).

1.6.4. Control de errores

Vamos a ver cómo TCP resuelve algunas situaciones conflictivas.

Recepción fuera de orden

TCP controla la recepción de datos fuera de orden mediante confirmaciones (ACK) de la siguiente forma.

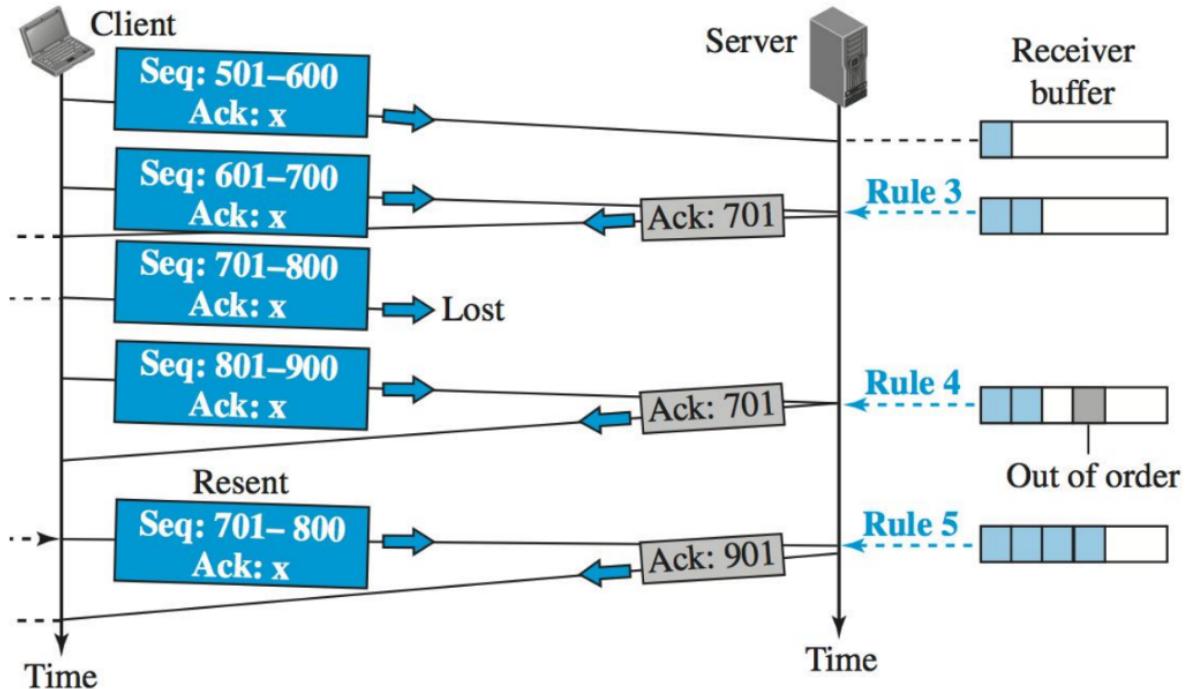


Figura 1.5: Recepción fuera de orden.

- Si hay huecos, el receptor envía el ACK del primer hueco que tenga en el buffer.
- Si no hay huecos envía el ACK del siguiente segmento esperado.
- Los segmentos duplicados se confirman para prevenir pérdidas de ACK.

Pérdida de un segmento

¿Qué hacer cuando se pierde un segmento, o se recibe erróneamente? para ello TCP tiene dos **mecanismos de retransmisión**:

1. **Temporizador de retransmisión:** si el emisor tiene segmentos sin confirmar, los reenvía al finalizar el temporizador. Si hay varios segmentos sin confirmar, el emisor reenvía el primero de la ventana.

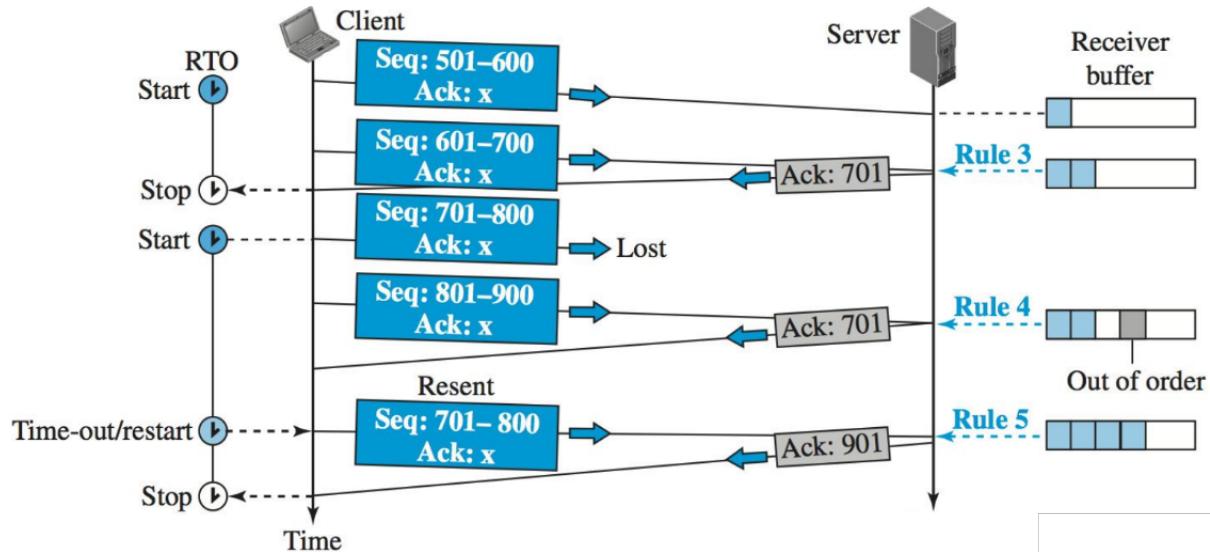


Figura 1.6: Temporizador de retransmisión.

2. Retransmisión rápida: si se reciben tres ACKs duplicados, el emisor reenvía el paquete.

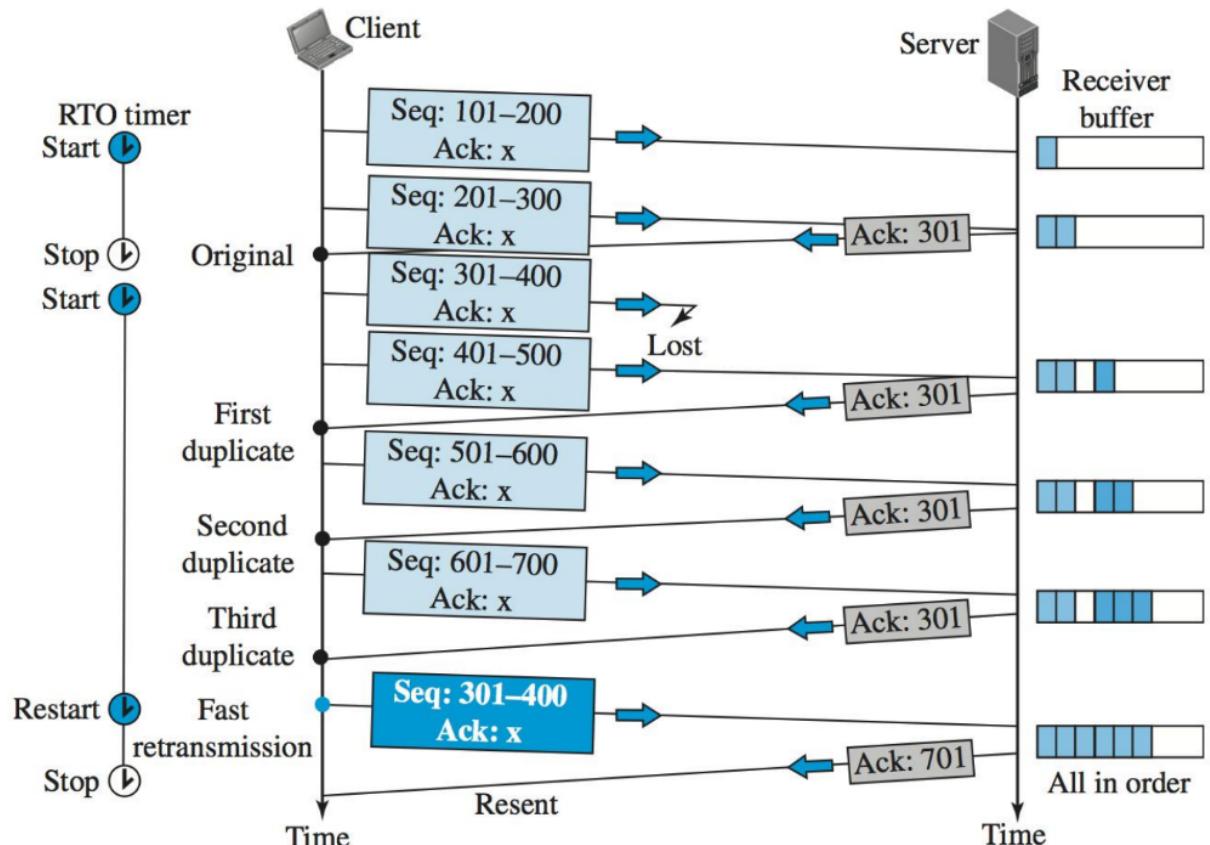


Figura 1.7: Retransmisión rápida.

Pérdida de un ACK

Con respecto a la pérdida de ACKs se pueden dar dos situaciones.

- **No expira el temporizador de retransmisión:** el emisor sigue enviando paquetes, por lo que si el siguiente ACK confirma un segmento posterior, también confirma todos los anteriores.

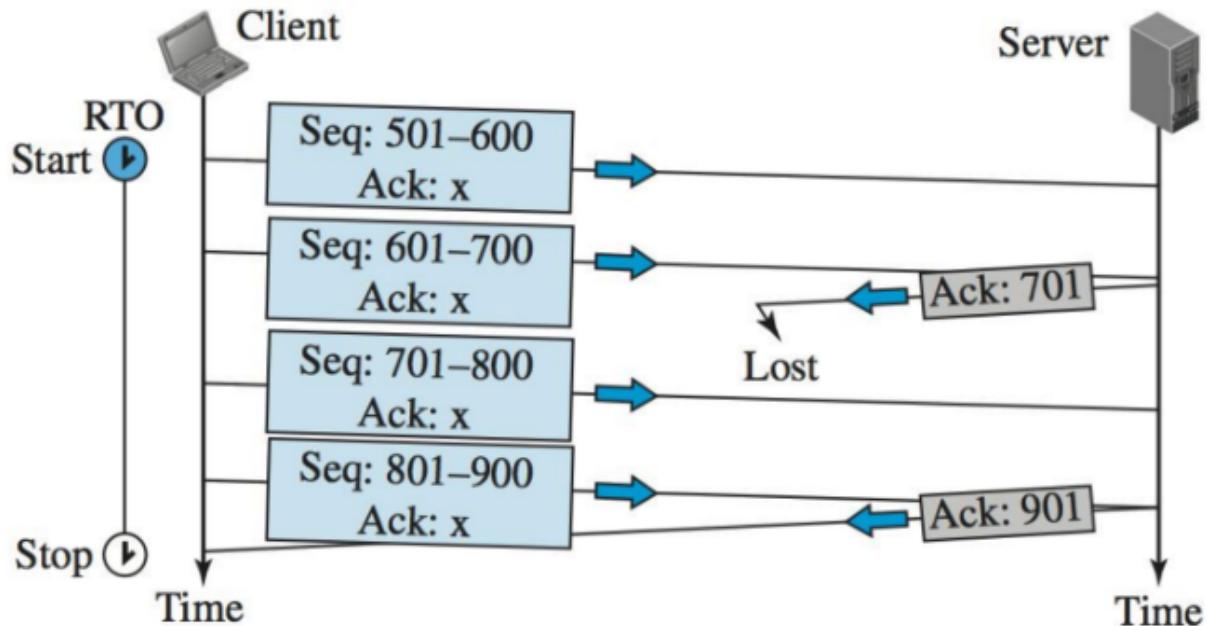


Figura 1.8: No expira el temporizador de retransmisión.

- **Expira el temporizador de retransmisión:** si el emisor no sigue enviando paquetes, cuando su temporizador de retransmisión expire, reenviará el paquete sin confirmar.

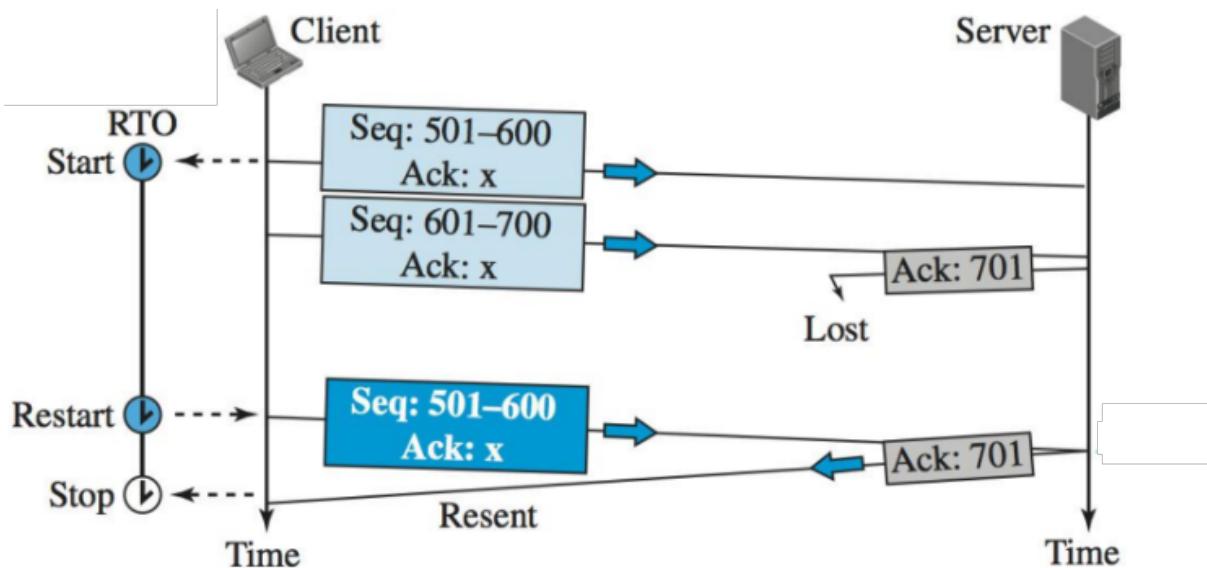
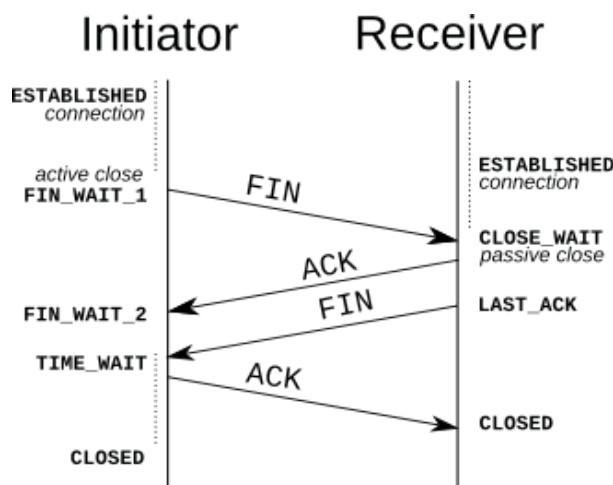


Figura 1.9: Expira el temporizador de retransmisión.

1.6.5. Temporizadores TCP

TCP utiliza 4 temporizadores.

- **Keepalive:** se utiliza para comprobar si un enlace está o no operativo. Cuando el temporizador expira (más de 2 horas, normalmente) envía una serie de señales *keepalive*. Si no recibe los ACK correspondientes, cierra la conexión.
- **TIMEWAIT:** en el cierre de conexión es el tiempo que espera la máquina que ha iniciado el cierre después de recibir el FIN (tanto en 3 como en 4 vías), para dar tiempo a la otra máquina a recibir el último ACK.



- **Temporizador de persistencia:** cuando el receptor advierte de un tamaño de ventana igual a 0, el emisor deja de enviar datos e inicia el temporizador de persistencia. El temporizador de persistencia se usa para proteger a TCP de una situación de interbloqueo que podría surgir si se pierde una actualización del tamaño de la ventana del receptor, y el emisor no puede enviar más datos hasta que reciba una nueva actualización del tamaño de la ventana del receptor. Cuando el temporizador expira, el emisor envía un pequeño paquete esperando un ACK, que le indique el nuevo tamaño de ventana.
- **Temporizador de retransmisión:** se utiliza cuando se espera un ACK del otro extremo. Veámoslo con más detalle.

Temporizador de retransmisión (TIMEOUT)

El tiempo de TIMEOUT se establece dinámicamente en función del tráfico de la red, y se pueden usar 3 algoritmos distintos para establecerlo.

El tiempo de ida y vuelta (RTT), es el tiempo transcurrido desde que se envía el segmento hasta que se recibe el ACK. El RTT puede presentarse de tres formas distintas:

- **RTT medido (RTTm):** es el RTT tal cual, por lo que puede experimentar grandes fluctuaciones. Por ejemplo si recibe un ACK acumulado, el RTTm se disparará.
- **RTT suavizado (RTTs):** es la media ponderada entre el RTTm y el último RTTS calculado. De forma que el primer RTTs = RTTm, y los siguientes valores dependen del RTTm actual y del RTTs anterior.
- **RTT desviación (RTTd):** Considera la variación del tiempo de ida y vuelta, y se calcula a partir del RTTm y del RTTs.

En éstos valores se basan los 3 **algoritmos** que puede usar el temporizador de retransmisión.

- **Jacobson:** utiliza únicamente RTTs.
- **Jacobson/Karels:** mejora el anterior combinando RTTS y RTTD.
- **Karn:** se basa en el anterior se basa solo en ACKs de segmentos que fueron enviados solo una vez.

1.6.6. Control de flujo

El control de flujo se utiliza para evitar desbordar el buffer del receptor cuando transmitimos muy rápido demasiados datos. Para ello TCP adaptará dinámicamente la ventana del emisor, dependiendo del tamaño de la ventana del receptor (anunciado cada ACK).

Si el receptor se queda sin espacio, su ventana tendrá tamaño = 0, y el emisor dejará de enviar datos. El emisor volverá a enviar datos de nuevo cuando el tamaño de la ventana del receptor vuelva a ser mayor que 0. Para saber el tamaño, utiliza el temporizador de persistencia.

Síndrome de la ventana tonta

Ocurre cuando o bien el servidor genera datos a un ritmo muy lento, o bien el cliente consume los datos a un ritmo muy lento.

Para evitar la ventana tonta en el **emisor**, se inventó el **algoritmo de Nagle**:

- Se envía el primer mensaje.
- Se espera a enviar los siguientes hasta que:
 1. se recibe un ACK del receptor.
 2. se acumulan x bytes.
 3. expira el TIMEOUT.

Para evitar la ventana tonta en el **receptor**, se inventó el **algoritmo de Clark**:

- Se anuncia tamaño de ventana 0 hasta que:
 1. Se puede recibir un segmento completo.
 2. Se ha liberado la mitad del buffer de recepción.

1.6.7. Control de la congestión

Cuando se pierde paquetes en Internet, la mayoría de las veces se debe a un problema de congestión en algún punto de la red: puede ocurrir que el router no pueda procesar paquetes al ritmo al que los recibe. Entonces, empieza a descartar paquetes.

El control de la congestión y el flujo son dos mecanismos diferentes. El control de flujo se encarga de ajustar la transmisión en el receptor, mientras que el **control de congestión** puede aparecer en cualquier punto de la red. Si un router no es capaz de gestionar todos los paquetes, empieza a descartar.

El emisor utiliza el ritmo de llegada de confirmaciones para regular el ritmo de envío de segmentos de datos. Para ello utiliza la Ventana de congestión (**CW**).

La ventana de congestión es complementaria a la ventana de recepción (**RW**) usada para el control de flujo.

- En una situación de no congestión (sin pérdida o retraso de segmentos) la ventana de congestión alcanza el mismo tamaño que la ventana de recepción ($CW = RW$).
- Cuando se produce una situación de congestión el tamaño de **CW** se va reduciendo progresivamente.
- Cuando la situación de congestión desaparece, el tamaño de **CW** se va aumentando progresivamente

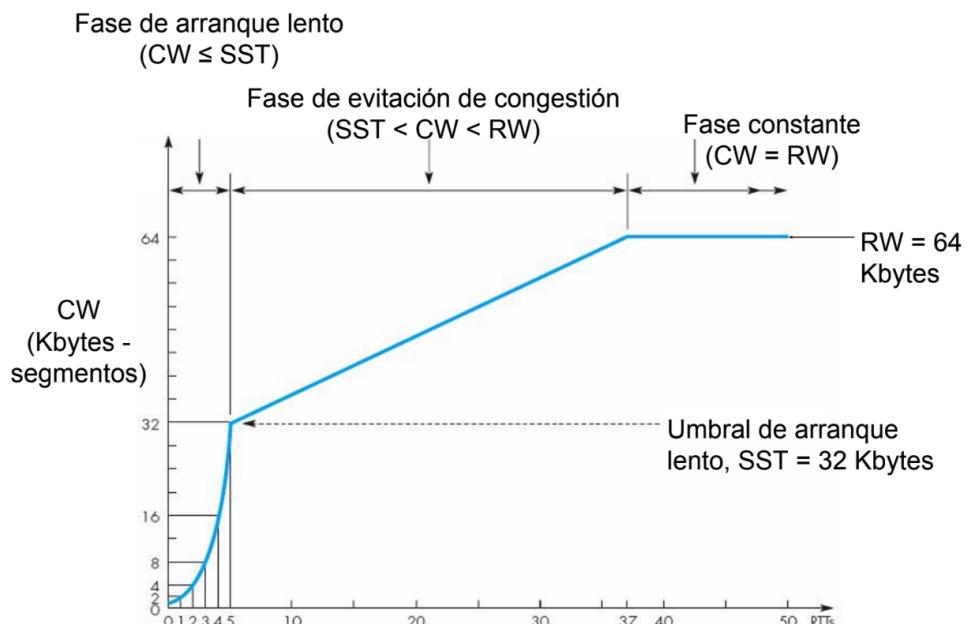
- El número máximo de bytes que puede enviar el emisor (**AW, Allowed Window**) es el mínimo de ambos tamaños de ventana:

$$AW = \min(RW, CW)$$

Fases del control de la congestión

La transmisión comienza con CW = 1.

1. **Fase de arranque lento:** el tamaño aumenta exponencialmente con cada segmento enviado hasta llegar a un umbral (SST, Slow Start Threshold). Inicialmente, el valor del SST suele ser de 64 Kbytes.
 2. **Fase de evitación de congestión:** Cuando se confirman todos los segmentos de la ventana la CW aumenta en +1 (crecimiento lineal).
 3. **Fase constante:** si la CW alcanza el mismo tamaño que la ventana de flujo (RW), el tamaño se mantiene.



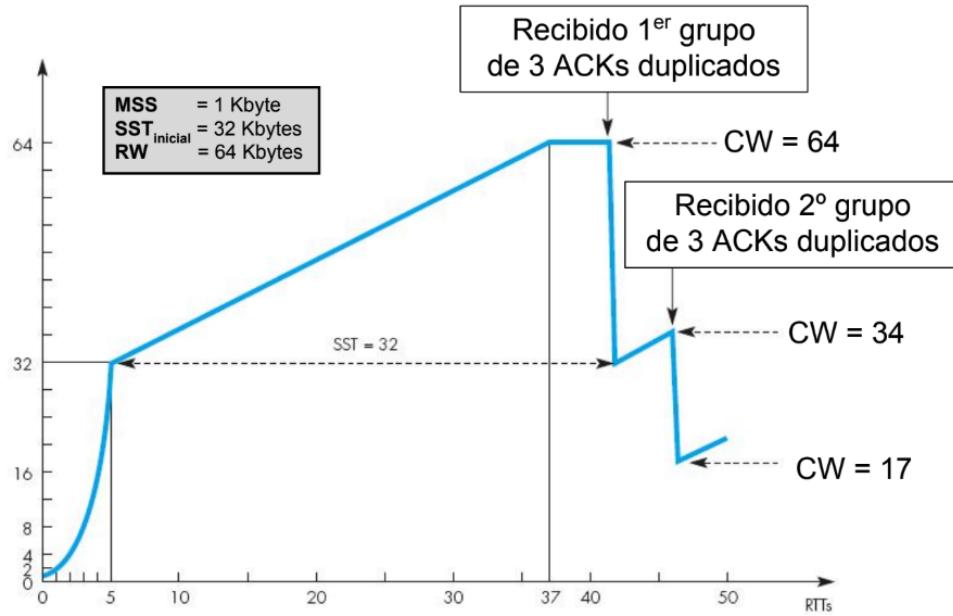
Mecanismos de control de la congestión

Durante la conexión pueden aparecer congestiones que se detectan de forma indirecta de dos formas distintas:

- Se reciben 3 ACKs duplicados. Esto indica un nivel de congestión leve, ya que sigue habiendo tráfico en la red.

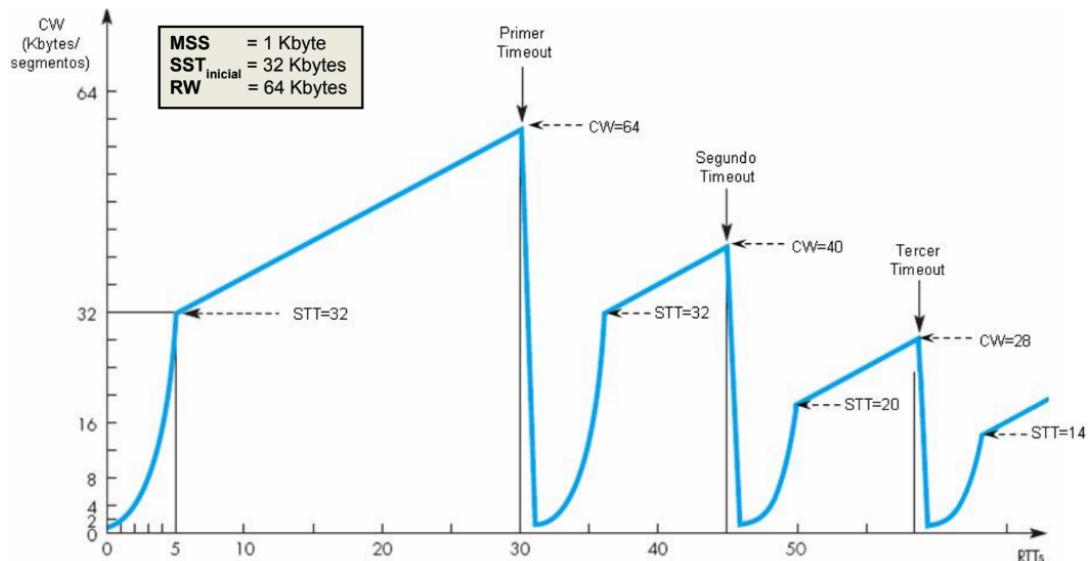
Ante esta situación se activa el método de recuperación rápida (fast recovery):

1. Se divide el valor de CW a la mitad.
 2. Se ejecuta el método de evitación de colisiones a partir de ese valor de CW.



- **Expira el temporizador** de retransmisión. Esto indica un nivel de congestión elevado, ya que se interpreta que el tráfico en la red está interrumpido. En este caso se realizan las siguientes acciones:

1. Se inicializa el tamaño de la ventana de congestión a CW = 1.
2. Se reduce el umbral de arranque lento (SST), fijándolo a la mitad del valor que tenía la CW antes de producirse el timeout.
3. Se ejecuta el método de arranque lento a partir de CW = 1.



1.7. Servicios de Red: Filtrado de paquetes.

1.7.1. Firewall y filtrado de paquetes.

Un **Firewall**: es un dispositivo configurado para permitir, limitar, cifrar o descifrar el tráfico de red en base a un conjunto de normas.

Tipos de Firewall.

- **En función del estado (stateless/stateful):** Si el filtrado se basa únicamente en la cabecera del paquete o además considera el estado de la conexión.
- **En función de la capa (de red o de aplicación):** consideran información no solo de las cabeceras si no de los datos(que normalmente pertenecen a protocolos de aplicación como http). Este tipo de firewalls se dice que realizan DPI (Deep packet inspection).

1.7.2. Iptables.

Netfilter es una herramienta del kernel de Linux que proporciona filtrado y modificación de paquetes.

El componente más popular construido sobre Netfilter es **iptables**, una herramienta de cortafuegos que permite no solamente filtrar paquetes, sino también realizar traducción de direcciones de red (NAT).

Para tener clara la estructura de iptables, primero debemos entender los siguientes conceptos:

- **Regla:** define qué hacer con un paquete que cumple unas determinadas características. *Por ejemplo descartar los paquetes con una IP destino determinada.*
- **Cadena:** es una lista ordenada de reglas que se aplican sobre los paquetes en distintos puntos del proceso del paquete.
- **Tabla:** es un conjunto de cadenas, cada una destinada a diferentes tipos de procesamiento sobre cada paquete.

Tablas predefinidas.

- **Tabla filter:** es la tabla por defecto. Contiene tres cadenas, que clasifican los paquetes:
 - INPUT: paquetes recibidos.
 - OUTPUT: paquetes generados.
 - FORWARD: paquetes que atraviesan el sistema.
- **Tabla NAT:** reescribe direcciones origen/destino y los puertos de un paquete. Contiene tres cadenas.
 - PREOUTING: se aplica a paquetes de entrada antes de pasar por la tabla de encaminamiento local.
 - POSTROUTING: se aplica a paquetes justo antes de ser enviados, modifica la dirección de origen de los paquetes.
 - OUTPUT: similar a la cadena homónima de la tabla filter.
- **Tabla Mangle:** sirve para cambiar algunos campos del paquete. Contiene las cinco cadenas anteriormente descritas.

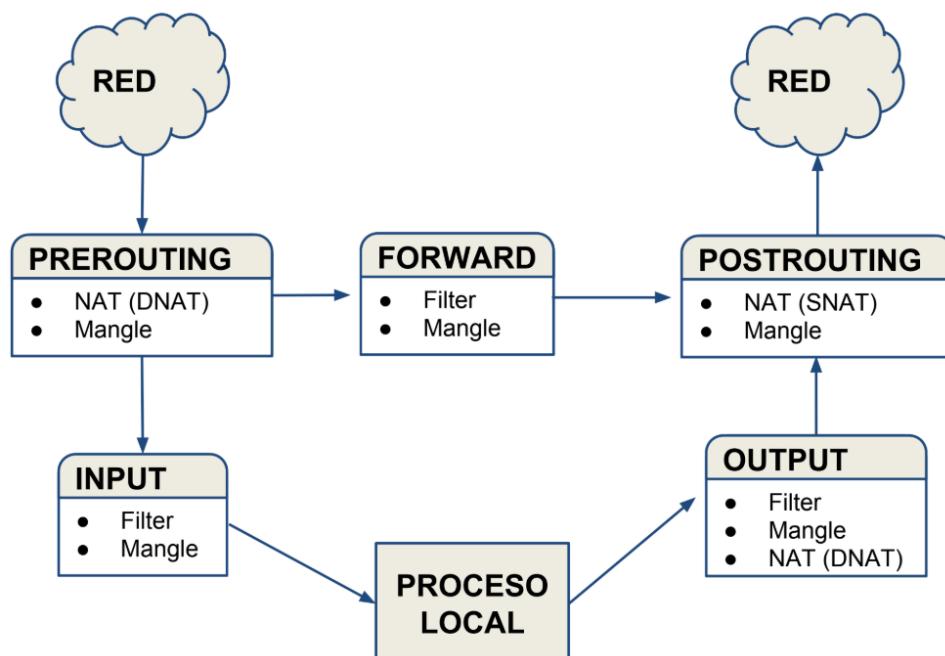


Figura 1.10: Diagrama de cadenas

Definición de Reglas.

Las reglas se pueden definir según la información del paquete o según el estado de la conexión. Debe incluir la **cadena** a la que se añade la regla y un **objetivo** (qué hacer si el paquete coincide).

Política por defecto	
-P INPUT	
-P OUTPUT	Política por defecto
-P FORWARD	

Cadena	
-A INPUT	Añade regla a cadena de entrada
-A OUTPUT	Añade regla a cadena de salida
-A FORWARD	Añade regla a la cadena forward (sólo en caso de routers)

Objetivo	
-j DROP	
-j ACCEPT	
-j REJECT	igual que DROP pero envía un ICMP de un tipo que puede definirse
-j LOG	

Opciones	
Consultar el manual de iptables	

```
# Establecer política por defecto para cadenas INPUT, OUTPUT y FORWARD
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
# Dejar entrar o salir cualquier paquete correspondiente a
# conexiones establecidas o relacionadas
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
# Permitir conexiones entrantes SSH (tcp/22) desde pc-oficina
iptables -A INPUT -s 200.1.1.1 -p tcp --dport 22 -m state \
--state NEW -j ACCEPT
# Permitir conexiones web salientes (tcp/80) a cualquier destino
iptables -A OUTPUT -p tcp --dport 80 -m state --state NEW -j ACCEPT
# Permitir conexiones pop3 salientes (tcp/110) con servidor de correo
iptables -A OUTPUT -d 22.1.1.1 -p tcp --dport 110 -m state \
--state NEW -j ACCEPT
# Permitir conexiones DNS salientes (udp/53) con servidor DNS
iptables -A OUTPUT -d 22.1.1.2 -p udp --dport 53 -m state \
--state NEW -j ACCEPT
```

Figura 1.11: Ejemplos de reglas

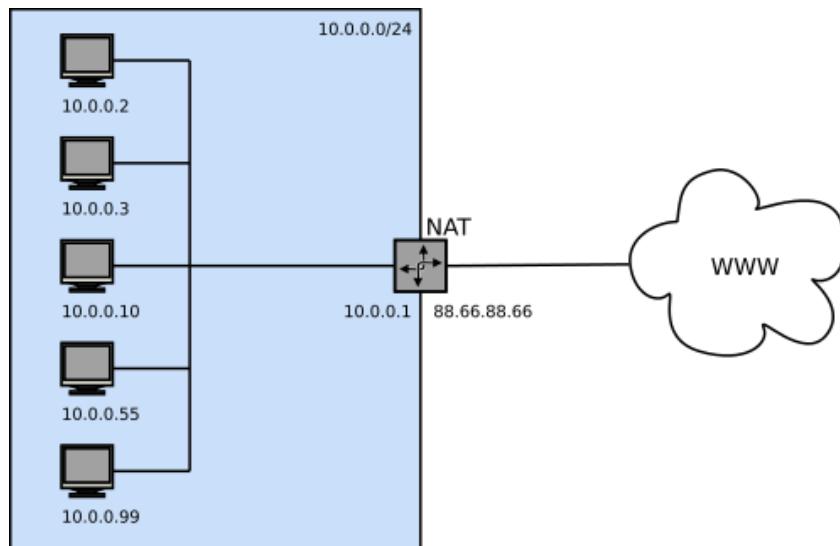
NAT

NAT: Network Address Translation: permite dar acceso a Internet a máquinas en redes privadas realizando traducciones.

NAT almacena una **tabla** que relaciona las direcciones privadas con las direcciones públicas.

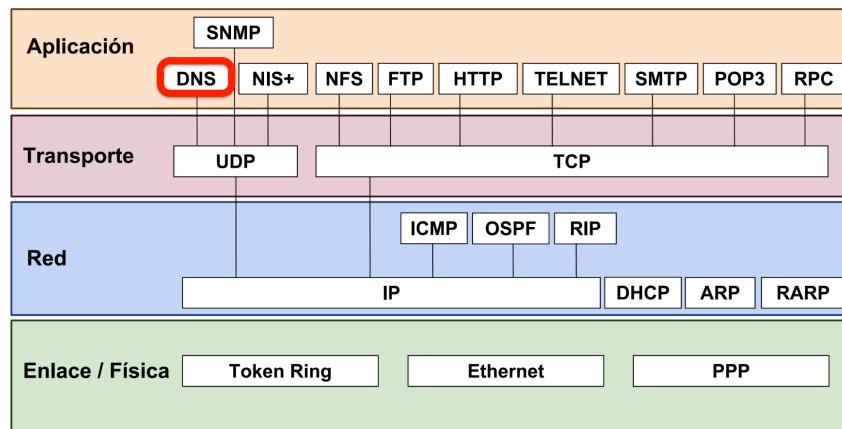
Existen los siguientes modos de funcionamiento:

- **Estáticas:** consiste en una asignación fija de N direcciones privadas a N direcciones públicas.
- **Dinámicas:** consiste en una asignación de N direcciones privadas a M direcciones públicas, donde ($M < N$). Por tanto las N direcciones privadas se disputan las M direcciones públicas.
- **Sobrecarga o NAPT:** asigna N direcciones privadas a 1 dirección pública.



- Para paquetes de salida se utiliza la técnica de **Masquerading**: traduce la dirección privada a la dirección pública asignada al router.
- Para los paquetes entrantes se utiliza la técnica de **Port forwarding, o servidores virtuales**, el router tiene una tabla con unas traducciones que depende del **número de puerto** asignado con el router.

1.8. DNS



DNS (Domain Name System), es un sistema de nomenclatura para dispositivos conectados a redes IP. Su función más importante es “traducir” nombres inteligibles para las personas en identificadores binarios asociados con los equipos, con el propósito de poder localizar y direccionar estos equipos mundialmente.

El servidor DNS utiliza una **base de datos distribuida** que almacena información asociada a nombres de dominio. DNS también define un protocolo de red.

1.8.1. Zonas y dominios.

Un nombre de dominio usualmente consiste en dos o más etiquetas, separadas por puntos. *Por ejemplo, www.ucm.es.*

$$(\text{root}) \rightarrow (\text{es}) \rightarrow (\text{ucm})$$

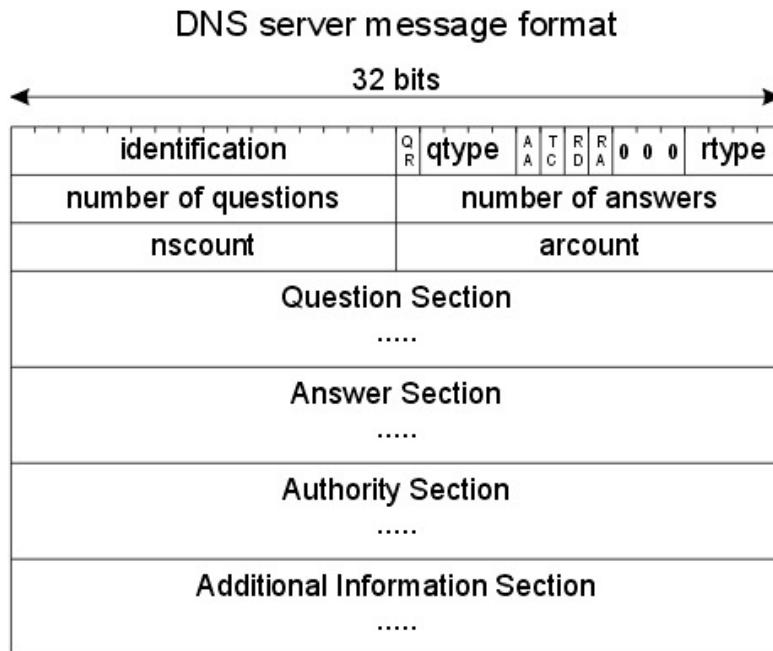
“es” es un subdominio de primer nivel, que pueden ser generales (com, gov, net..) o de país (fr, es, uk...).

“ucm” es un subdominio de segundo orden, etc...

Una **zona** DNS es una parte del espacio de nombres que se delega a una entidad legal. Cuando un navegador web necesita encontrar la dirección IP para un nombre de host, realiza una búsqueda de DNS en el servidor DNS que administra la zona para ese nombre de host.

Un **FQDN** (fully qualified domain name) es un nombre que incluye el nombre de la computadora y el nombre de dominio asociado a ese equipo. Por ejemplo, dada la computadora llamada “serv1” y el nombre de dominio “bar.com.”, el FQDN será “serv1.bar.com.”

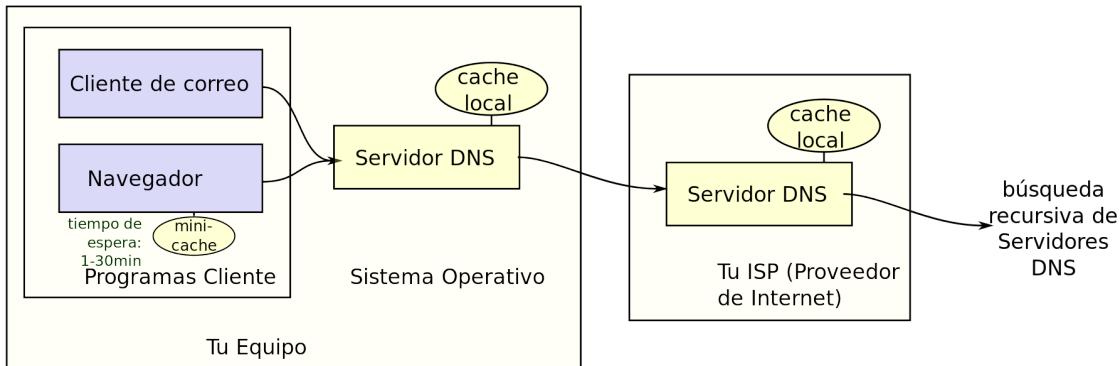
1.8.2. Datagrama DNS



- La **Cabecera** se divide en 6 campos, que permiten identificar el datagrama y determinar su contenido.
- Las secciones **Pregunta** y **Respuesta** incluye el nombre de dominio y el tipo de registro por el que se pregunta.
- Sección de **Autoridad** especifica los servidores de nombre oficiales de la zona por la que estamos preguntando.
- La sección **Adicional** incluye registros que pueden ser de ayuda (resolver)

1.8.3. Características del protocolo DNS

1. La base de datos DNS se estructura en **registros**, que son la información básica que se intercambia y cachea en los servidores. DNS gestiona diferentes tipos de registros para almacenar servidores de nombres, asignaciones nombre-IP, etc...
2. **BIND** es la implementación DNS más usada y es Open Source.
3. Los datagramas DNS se encapsulan en **datagramas UDP**, aunque se puede utilizar TCP en algunos casos (como obtener toda la información de un servidor, o si la respuesta tiene más de 512 bytes)
4. Ante una consulta DNS, se busca en primer lugar si dispone de la respuesta en la memoria caché. En caso contrario, se inicia la búsqueda de manera recursiva. Una vez encontrada la respuesta, el servidor DNS guardará el resultado en su memoria caché para futuros usos y devuelve el resultado.



5. **Cachear** la resolución de direcciones mejora notablemente la eficiencia. La relación nombre-IP es prácticamente estática, por lo que las respuestas se cachean durante TTL ("time-to-live") que varía según el nivel al que se encuentren, por ejemplo:

- Servidores de la zona ".es", 2 días.
- Servidores del dominio "rediris.es", 1 día.
- IP de "www.rediris.es", 2 horas.

6. Existen los siguientes **Tipos de servidores**:

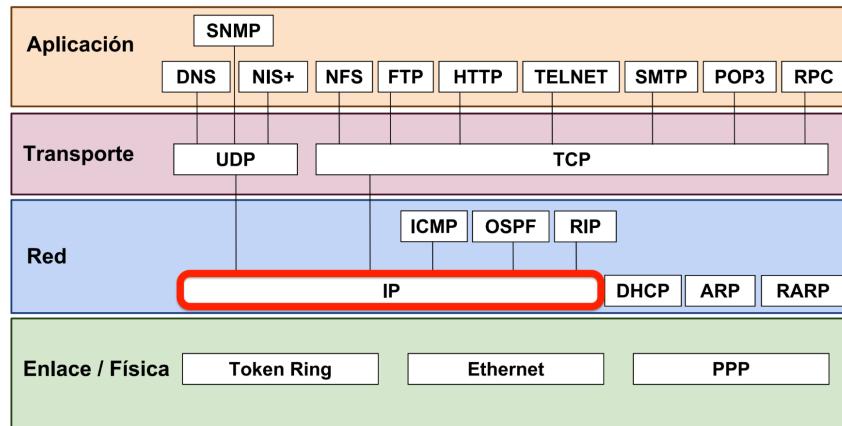
- **Autoritativos:** Representan oficialmente a la zona.
 - Primarios o maestros: tienen la copia oficial en disco de la BD.
 - Secundarios o esclavos: obtienen la BD de los primarios (zone transfer).
- **De cache:** Guardan los resultados de las búsquedas realizadas. No tienen ningún registro DNS propio, ni son autoritativos para ninguna zona.
- **No recursivos:** cuando no disponen el registro de la consulta devuelven una referencia al servidor de nombres que puede tenerlo. Los servidores autoritativos suelen ser no recursivos.
- **Recursivos:** resuelven cada referencia hasta devolver la respuesta al cliente. En la configuración de los clientes deben usarse servidores recursivos.

7. La **base de datos DNS** consiste en archivos de texto mantenidos en el servidor primario de la zona. La base de datos se compone de los siguientes registros:

- **SOA:** marca el comienzo de definición de una zona.
- **NS:** especifica los servidores autoritativos para la zona e incluye los servidores de nombres de los subdominios delegados a otras organizaciones.
- **A:** El registro Address (A para IPv4 y AAAA para IPv6) es la base de DNS. Incluye la traducción directa (nombre → IP)
- **PTR:** El registro Pointer contiene la traducción inversa (IP → nombre)
- **MX:** es usado por los sistemas de correo para encaminar los mensajes eficientemente. Permite recibir de forma centralizada el correo de una organización.

- **CNAME:** permite definir un alias para el nombre canónico. Deben siempre apuntar a un dominio (nunca a una IP).
8. El comando **dig** se puede usar para realizar consultas DNS.

1.9. Protocolo IPv6



IPv6 es una versión del protocolo IP, creada como respuesta a las limitaciones de IPv4. Algunas de éstas limitaciones son:

- El agotamiento de direcciones.
- El formato complejo de la cabecera.
- Problemas de seguridad.

A pesar de que IPv6 es usable desde hace años, a día de hoy IPv4 es el protocolo dominante en internet. Por ello, éstas limitaciones se han ido solucionando con parches durante los últimos años. *Por ejemplo se ha adaptado la seguridad de IPv6 a IPv4.*

1.9.1. IPv4 vs IPv6

Frente a IPv4, IPv6 presenta algunas ventajas:

- El espacio de direcciones es mucho mayor (128 bits, comparado con los 32 bits de IPv4).
- El formato de cabecera se simplifica, con lo que los routers son capaces de procesar más rápido.
- No se necesita DHCP, IPv6 es capaz de hacer su trabajo.
- IPv6 elimina las direcciones broadcast y las implementa como un caso especial de multicast.
- Proporciona opciones de seguridad tanto para autenticación como para cifrado.
- Implementa las direcciones Anycast, que envía el datagrama a la máquina más cercana.
- Da soporte para tráfico en tiempo real. *Por ejemplo: VoIP.*

1.9.2. Direcciones IPv6

IPv4	IPv6
Unicast	Unicast
Multicast	Multicast
Broadcast	Anycast

Un paquete dirigido a una dirección **unicast** se entregará únicamente al interfaz identificado con dicha dirección IP.

Las direcciones **multicast** identifican a un grupo de interfaces. Un paquete dirigido a una dirección multicast se entrega a todos los interfaces identificados con esa dirección.

IPv6 no implementa broadcast, el mismo efecto puede lograrse enviando un paquete al grupo de multicast de enlace-local todos los nodos (all hosts). Por lo tanto, no existe el concepto de una dirección de broadcast y así la dirección más alta de la red (la dirección de broadcast en una red IPv4) es considerada una dirección normal en IPv6.

Las direcciones **anycast** son asignadas del espacio de direcciones unicast. De esta forma, las direcciones anycast no se pueden distinguir sintácticamente de las unicast. Cuando una dirección unicast es asignada a más de una interfaz, esta se convierte en una dirección anycast.

Notación de direcciones.

IPv6 usa la notación **hexadecimal**.

- La dirección se divide en 8 grupos de 16 bits.
- Cada grupo se escribe en hexadecimal con 4 dígitos.
- Los grupos se separan por ":".
- La longitud del prefijo se denota en CIDR.
- Existen algunos prefijos reservados, por ejemplo para direcciones multicast o ULA.
- Los 64 bits menos significativos determinan el ID del interfaz (distinto del host), que permite relacionar la dirección IP de red con la dirección MAC.

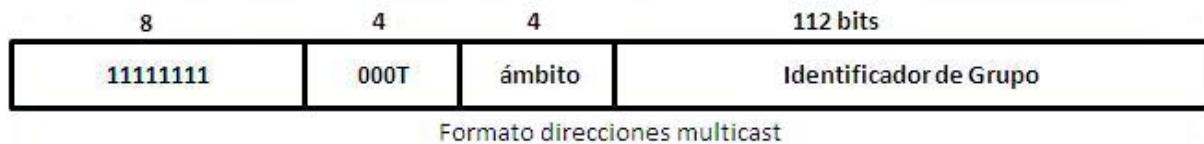
Como se generan direcciones muy largas, se usan algunos métodos para abreviar.

- Omitir los ceros a la izquierda de cada grupo.
- Las cadenas de ceros seguidos se pueden comprimir con el símbolo "::". Aunque este símbolo solo puede aparecer una vez.

Ejemplo: FE80::8:800:200C:741A/64

Direcciones Multicast

Una dirección multicast IPv6 es un identificador para un grupo de interfaces (normalmente en diferentes nodos). Una interfaz puede pertenecer a cualquier número de grupos multicast. Las direcciones multicast tienen el siguiente formato:



Las direcciones multicast comienzan con el prefijo **FF::/8**.

Direcciones públicas y privadas

Las direcciones **Global Unicast** en IPv6 son el equivalente de las direcciones IP públicas en IPv4. Estas direcciones IP pueden ser encaminadas a través de la Internet. Actualmente usan el prefijo **2000::/3**, que permite 2^{45} sitios diferentes.

De igual modo que IPv4 dispone de direcciones **reservadas** para uso privado (*Por ejemplo 192.168.x.x*), IPv6 dispone de **ULA (Unique Local Addresses)**, que son todas las direcciones que comienzan con el prefijo **fc00::/7**.

Ámbitos y zonas

Las direcciones IPv6 incluyen un **ámbito**, que indican en qué parte de la red la dirección es **válida**.

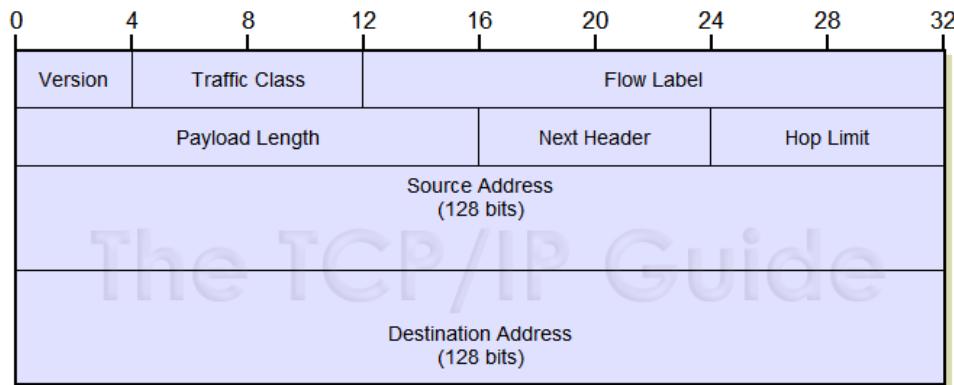
Una **zona** es una región conectada de la red de un ámbito determinado, dentro de la cual se garantiza que todas las direcciones son **únicas**. Dentro de una zona, los datagramas no se redirigen a una zona distinta, aunque sea del mismo ámbito. Para indicar la zona de una dirección se usa el símbolo “%”.

Ejemplo: FE80::1234 %1

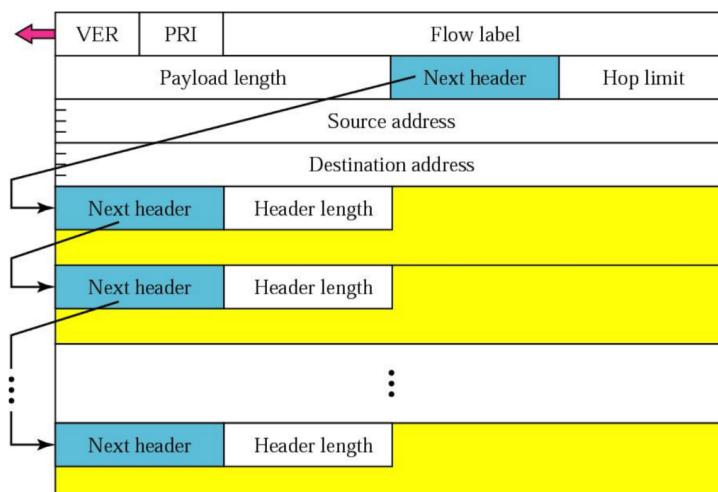
1.9.3. Datagrama IPv6

Un datagrama IPv6 está compuesto principalmente de dos partes: la cabecera y los datos.

Cabecera



- **Traffic Class:** distingue diferentes requisitos de entrega del datagrama, es similar al campo DS/ToS de IPv4.
- **Flow Label:** Etiqueta el paquete como perteneciente a un **flujo** para mejorar el procesamiento realizado por los encaminadores de la red. Un flujo es un conjunto de paquetes que comparte las mismas características (origen/destino, requisitos, etc...).
- **Payload Length:** longitud sin contar la cabecera (máx. 64 Kbytes).
- **Hop Limit:** similar al campo TTL de IPv4.
- **Next Header:** Define la siguiente cabecera en el datagrama (dentro de la sección de datos), puede ser:
 - Una cabecera de extensión, similar al campo opciones de IPv4.
 - un protocolo de nivel superior, encapsulado en la sección de datos (2=ICMP, 6=TCP, 17=UDP...)

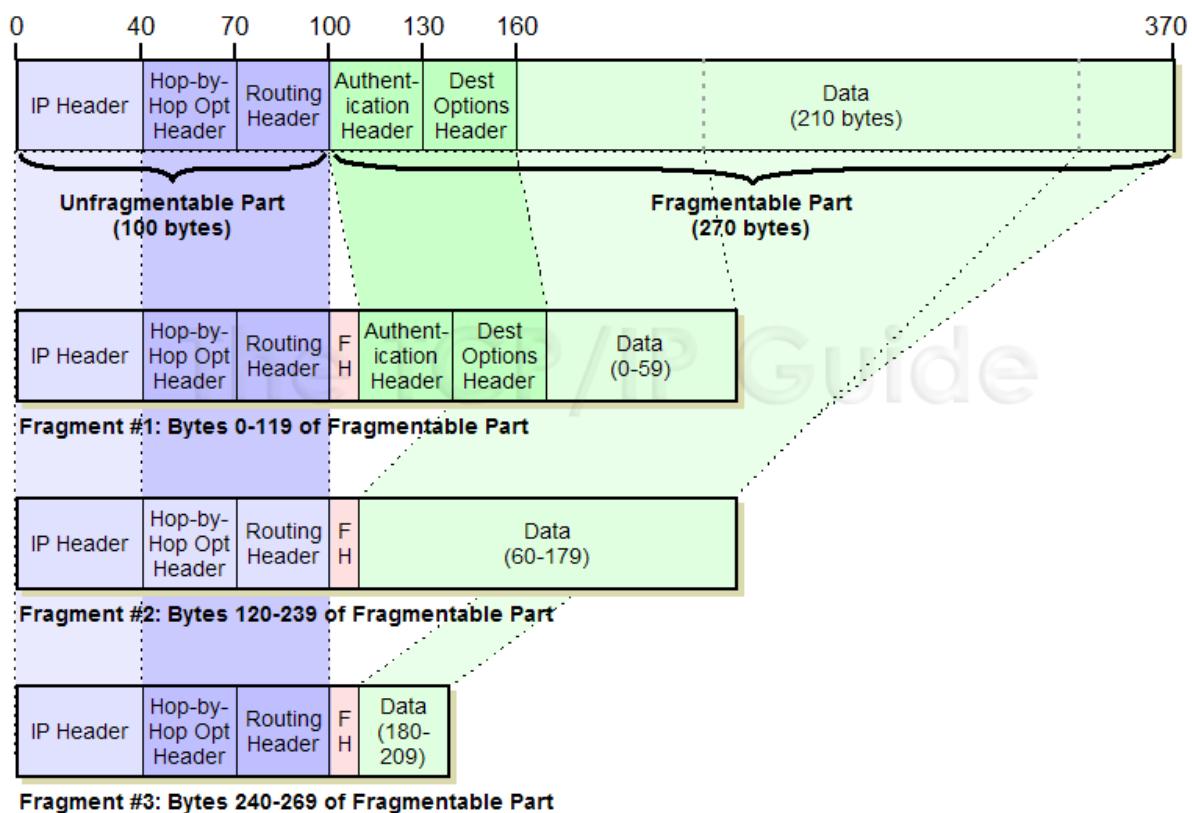


Cambios con respecto a IPv6

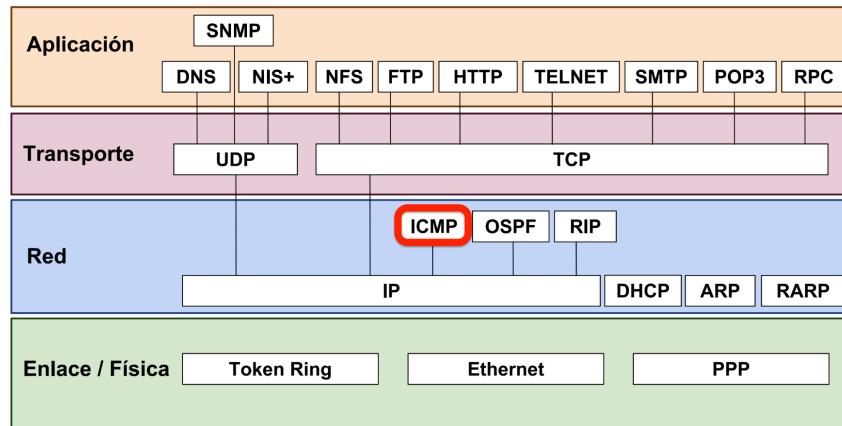
- El campo Header Length se ha eliminado, ya que la longitud es fija.
- No hay campo Checksum, ya que se realiza por los protocolos superiores.
- Los campos de fragmentación se eliminan de la cabecera y se implementan en cabeceras de extensión.

Fragmentación

Los routers IPv6 no hacen fragmentación. Los nodos IPv6 requieren ya sea hacer descubrimiento de MTU, realizar fragmentación extremo a extremo o enviar paquetes del tamaño mínimo MTU para IPv6 (1280 bytes).



1.10. Protocolo ICMPv6

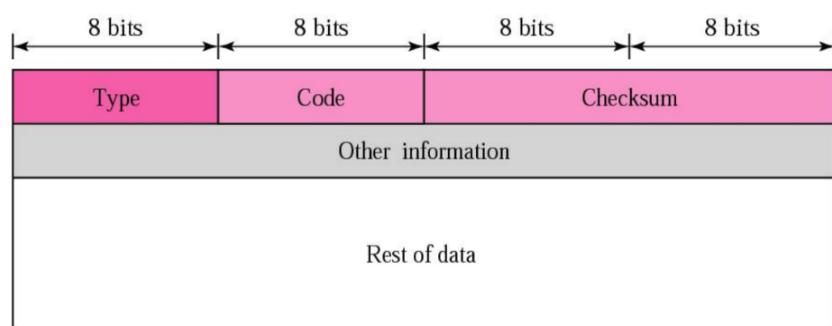


ICMPv6 (Internet Control Message Protocol version 6) es una nueva versión de ICMP y una parte importante de la arquitectura IPv6.

ICMPv6 asume el papel de los siguientes protocolos auxiliares en IPv4:

- ICMPv4, protocolo de mensajes de control.
- IGMP, protocolo de gestión de grupos multicast.
- ARP, protocolo de resolución de direcciones.

Todos los mensajes ICMPv6 tienen un formato común:



1.10.1. Mensajes ICMPv6

Mensajes de error

Los mensajes de error se pueden producir cuando:

- El destino es inalcanzable.
- El datagrama es demasiado grande.
- Tiempo excedido.
- Problema de parámetros.

Mensajes de información

Proporcionan información de diagnóstico, pueden ser:

- Echo request.
- Echo reply.

Descubrimiento de vecinos

Neighbor Discovery (ND) es un tipo de mensaje equivalente al protocolo Address Resolution Protocol (ARP) en IPv4.

Permite que los terminales aprendan las direcciones IPv6 de los vecinos y opera sobre hosts y routers en el mismo enlace.

Se pueden realizar las siguientes funciones:

- **Descubrimiento de vecino.** Se utilizan dos tipos de mensajes:
 1. Neighbor Solicitation. Este mensaje se genera para:
 - Averiguar la dirección física asociada a una dirección IP (como ARP request en IPv4).
 - Determinar si un nodo vecino sigue siendo alcanzable.
 - Detectar si la dirección IP está duplicada.
 2. Neighbor Advertisement. Este mensaje se genera para:
 - Responder a un mensaje de solicitud de vecino (como ARP reply en IPv4).
 - Anunciar un cambio en la dirección física de un interfaz.
- **Descubrimiento de router.** Se utilizan dos tipos de mensajes:
 1. Router Solicitation: este mensaje se genera tras la activación de un interfaz, para detectar los encaminadores y realizar la autoconfiguración del interfaz.
 2. Router Advertisement: los envían los routers para anunciar su presencia en la red. Son enviados de forma periódica y como respuesta a mensajes Router Solicitation.
- **Redirección.** Permite notificar a un host una ruta más adecuada para alcanzar un determinado destino.

1.11. Encaminamiento en Internet

En una red, el encaminamiento consiste en encontrar un camino, desde el origen al destino, a través de nodos de conmutación o routers intermedios. También es necesario decidir cuál es el mejor camino posible (camino más corto) según alguna métrica (*número de nodos, distancia geográfica, retardo, etc...*).

Cuando un encaminador recibe un paquete lo retransmite por el enlace adecuado para alcanzar el destino. Esta elección la realiza según:

- **Tablas de encaminamiento.** Guardan una lista de las direcciones IP y el siguiente salto que tienen asociado.
- **Etiquetas.** Cada datagrama IP se etiqueta y se conmuta según una etiqueta (campo “Flow Label” en IPv6). Esto reduce la complejidad de la tabla de encaminamiento.

1.11.1. Técnicas de Encaminamiento

- **Encaminamiento local:** no tiene en cuenta la topología de la red y usa únicamente información local. Las técnicas más comunes son: Encaminamiento aleatorio, Encaminamiento aislado, e Inundación.
- **Encaminamiento estático:** Las tablas de encaminamiento se construyen manualmente y no se adaptan a los cambios de la red.
- **Encaminamiento dinámico:** las tablas de encaminamiento se construyen de forma automática, mediante el intercambio periódico de información entre los encaminadores. Las técnicas más comunes son:
 - Encaminamiento por **vector de distancias** (*ejemplo: RIP*).
 - Encaminamiento por **estado de los enlaces** (*ejemplo: OSPF*).
 - Encaminamiento por **vector de rutas** (*ejemplo: BGP*).

Sistemas Autónomos (AS)

Un AS es una conjunto de redes y encaminadores gestionados y administrados por una misma autoridad. Se componen de:

- **Routers Internos:** interconectan redes dentro del propio AS. Utilizan **protocolos internos (IGP)** como RIP o OSPF.
- **Routers Frontera:** interconectan varios ASs. Utilizan **protocolos externos (EGP)** como BGP.

1.12. Vector de distancias - Protocolo RIP

El **vector de distancias** es un método de enrutamiento que utiliza el algoritmo de Bellman-Ford para calcular las rutas.

1.12.1. Funcionamiento

Imaginemos una red con la siguiente topología:

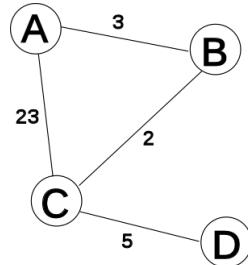


Figura 1.12: Consta de 4 nodos. El número encima de cada arista indica la distancia del enlace.

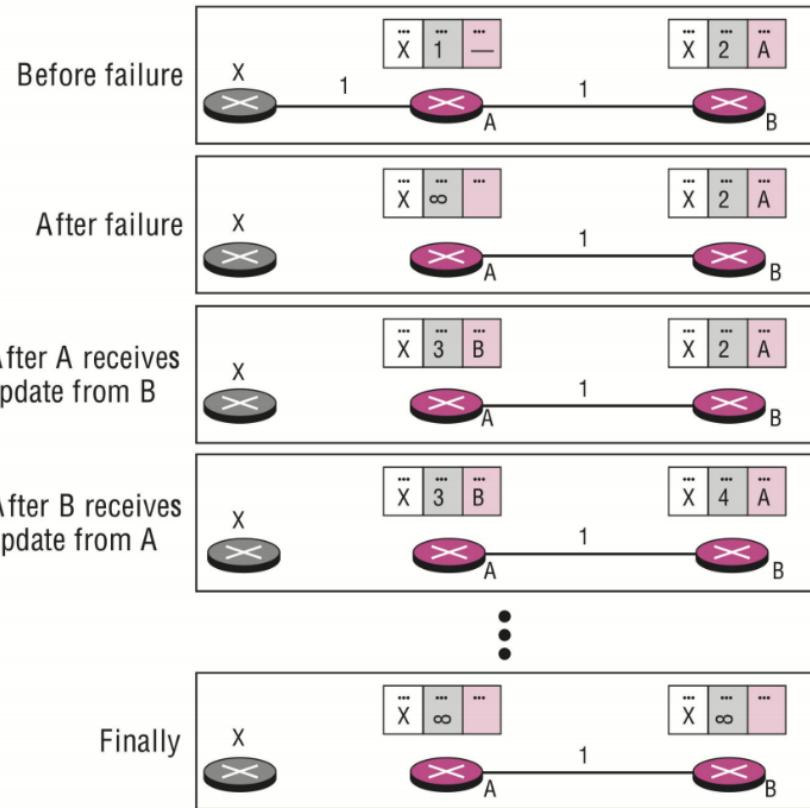
Inicialmente los routers sólo conocen sus rutas directas. Vemos como en cuatro unidades de tiempo (T), todos los routers disponen de todas las distancias posibles.

	desde A	vía A	vía B	vía C	vía D	desde B	vía A	vía B	vía C	vía D	desde C	vía A	vía B	vía C	vía D	desde D	vía A	vía B	vía C	vía D
T=0	a A					a A	3				a A	23				a A				
	a B		3			a B					a B		2			a B				
	a C			23		a C			2		a C					a C			5	
	a D					a D					a D				5	a D				
	desde A	vía A	vía B	vía C	vía D	desde B	vía A	vía B	vía C	vía D	desde C	vía A	vía B	vía C	vía D	desde D	vía A	vía B	vía C	vía D
T=1	a A					a A	3		25		a A	23	5			a A			28	
	a B		3	25		a B					a B	26	2			a B		7		
	a C	5	23			a C	26		2		a C					a C		5		
	a D		28			a D		7			a D				5	a D				
	desde A	vía A	vía B	vía C	vía D	desde B	vía A	vía B	vía C	vía D	desde C	vía A	vía B	vía C	vía D	desde D	vía A	vía B	vía C	vía D
T=2	a A					a A	3		25		a A	23	5	33		a A			10	
	a B		3	25		a B					a B	26	2	12		a B		7		
	a C	5	23			a C	26		2		a C					a C		5		
	a D		10	28		a D	31		7		a D	33	9		5	a D				
	desde A	vía A	vía B	vía C	vía D	desde B	vía A	vía B	vía C	vía D	desde C	vía A	vía B	vía C	vía D	desde D	vía A	vía B	vía C	vía D
T=3	a A					a A	3		7		a A	23	5	15		a A			10	
	a B		3	25		a B					a B	26	2	12		a B		7		
	a C	5	23			a C	8		2		a C					a C		5		
	a D		10	28		a D	31		7		a D	33	9		5	a D				

Figura 1.13: El camino más corto está marcado con el color verde, un camino más corto nuevo está indicado en amarillo.

1.12.2. Problemas

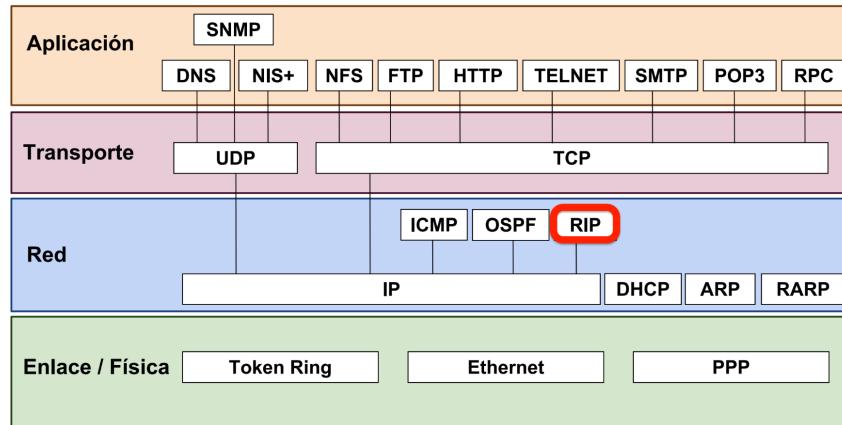
El algoritmo Bellman-Ford utilizado en Vector de Distancias no previene de la aparición de bucles. El problema de la **cuenta a infinito** es que hace que los costes o distancias se incrementen indefinidamente sin que el algoritmo llegue a converger nunca.



Soluciones a la cuenta a infinito:

- **Establecer el infinito a un número pequeño.** Por ejemplo, en RIP el infinito se establece en 16 saltos.
- **Horizonte dividido:** los destinos aprendidos a través de un determinado enlace nunca se difunden a través de dicho enlace. *Ejemplo: El nodo B no enviará al nodo A información sobre el destino X.*
- **Horizonte dividido con ruta inversa envenenada:** los destinos aprendidos a través de un determinado enlace sí se difunden a través de dicho enlace, pero con distancia infinita. *Ejemplo: El nodo B anunciará al nodo A que el destino X está a distancia infinita.*
- **Actualizaciones forzadas:** cuando un encaminador detecta una modificación en su tabla de rutas inmediatamente difunde esta información a sus vecinos. De esta forma, los cambios en la topología se propagan de forma rápida a todos los puntos de la red.

1.12.3. Protocolo RIP



RIP (Routing Information Protocol), es un protocolo de puerta de enlace interna (Interior Gateway Protocol, IGP) utilizado por los routers para intercambiar información acerca de redes IP. Su algoritmo de encaminamiento está basado en el **vector de distancia**, ya que calcula la ruta más corta posible a partir del número de saltos.

Versiones

- RIP versión 1 (1993).
- RIP versión 2 (1998).
- RIPng para IPv6 (1997).

Mensajes

Los mensajes tienen una cabecera con el tipo de mensaje y la versión del protocolo RIP, y un máximo de 25 entradas.

Las entradas en **RIPv1** contienen la dirección IP de la red de destino y la métrica. Las entradas en **RIPv2** incluyen además su máscara y el siguiente encaminador.

Los mensajes RIP pueden ser de dos tipos:

- **Petición (REQUEST):** enviados por algún encaminador recientemente iniciado que solicita información de los encaminadores vecinos.
- **Respuesta: (RESPONSE)** mensajes con la actualización de las tablas de encaminamiento. Existen tres tipos:
 1. Periódicos: Se envían cada 30 segundos. Para indicar que el enlace y la ruta siguen activos. Se envía la tabla de encaminado completa.
 2. Mensajes enviados como respuesta a mensajes de petición.
 3. Mensajes enviados cuando cambia algún coste. Se envía toda la tabla de encaminado.

Temporizadores

RIP utiliza distintos temporizadores:

- **Temporizador periódico:** controla la publicación de los mensajes periódicos. Se debe ajustar el temporizador a 30s.
- **Temporizador de caducidad:** establece cuánto tiempo puede estar una ruta en la tabla sin ser actualizada.
- **Temporizador de Recolección de Basura:** controla el tiempo que pasa entre que una ruta es invalidada (o marcada como inalcanzable) y el tiempo que pasa hasta que se elimina la entrada de la tabla de ruteo.

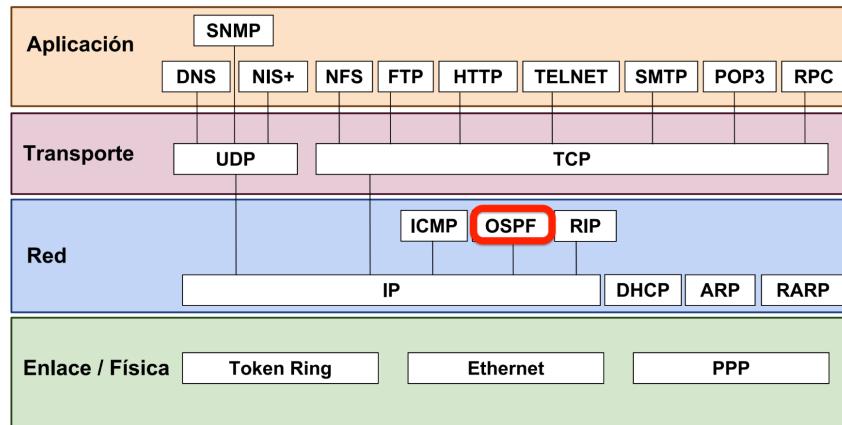
1.13. Estado de los enlaces - Protocolo OSPF

Estado de los enlaces consiste en que un router comunica al resto de nodos de la red cuáles son sus vecinos y a qué distancia está de ellos. Con la información que un nodo de la red recibe de todos los demás, puede construir un “mapa” de la red, y sobre él calcular los caminos óptimos. El encaminamiento por estado de enlace sustituyó al método de vector de distancias.

Funcionamiento. Lo podemos dividir en cinco pasos fundamentales:

1. Descubrir a sus vecinos y sus direcciones.
2. Medir el costo a cada uno de sus vecinos.
3. Construir el paquete con la información recabada.
4. Enviar este paquete al resto de routers.
5. Calcular la ruta mínima al resto de routers. Una vez que el router ha completado la recopilación de información, puede construir el grafo de la subred. De esta manera, se puede utilizar el algoritmo de Dijkstra para calcular el camino más corto a todos los nodos.

1.13.1. Protocolo OSPF



OSPF (Open Shortest Path First), es un protocolo de red para encaminamiento jerárquico de pasarela interior (IGP), que usa el algoritmo Dijkstra para calcular la ruta más corta entre dos nodos.

Su medida de métrica se denomina **cost**, y tiene en cuenta diversos parámetros tales como el ancho de banda y la congestión de los enlaces. OSPF es probablemente el protocolo IGP más utilizado en redes grandes.

Se desarrolló como alternativa a RIP para aliviar sus limitaciones:

- Distribuye la carga entre caminos equivalentes.
- Particionado lógico de la red para reducir la cantidad de información anunciada.
- Convergencia más rápida, propaga inmediatamente los cambios en las rutas.
- Soporte para máscaras de longitud variable (VLSM) y CIDR

Área

Es una agrupación lógica de encaminadores y redes, con un identificador de área de 32 bits (Area ID).

Los encaminadores mantienen únicamente información de su área y limitan el número de intercambios de información de los enlaces.

Tipos de router en OSPF

- **Intra-Area Routers (IA):** todos sus interfaces están en el área y mantiene sólo información de la topología de su área.
- **Area Border Routers (ABR):** conectado a dos o más áreas. Mantiene una Base de Datos para cada una de las áreas a las que está conectado.
- **AS Boundary Routers (ASBR):** Situado en la frontera del AS, intercambia rutas entre la red OSPF y otros sistemas.

Relación con los vecinos en OSPF

Cada router OSPF realiza un seguimiento de sus nodos vecinos. Para el descubrimiento de vecinos se utiliza el protocolo **OSPF Hello**.

OSPF utiliza paquetes **hello** y dos temporizadores para saber si un vecino sigue disponible. Cuando un router recibe un mensaje Hello de otro router que contiene su propio ID, establece una relación de vecindad con dicho router.

Después de establecer la vecindad se intercambian las base de datos con el estado de los enlaces. Con esa información construye su árbol de rutas. El árbol de rutas incluye tanto encaminadores (ID) como redes (IP) y el coste asociado.

1.14. Vector de Rutas - Protocolo BGP

Vector de Rutas es un protocolo de enrutamiento de red que mantiene la información de ruta actualizada dinámicamente. Está basado en el de vector de distancias.

Es diferente del enrutamiento por vector de distancia y el enrutamiento por estado de enlace. Cada entrada en la tabla de enrutamiento contiene la red de destino, el siguiente router y la ruta para llegar al destino. Utiliza CIDR y detecta bucles de forma sencilla.

1.14.1. Protocolo BGP

BGP (Border Gateway Protocol) es un protocolo de intercambio de información de encaminamiento entre sistemas autónomos. Es un ejemplo de protocolo EGP.

Los encaminadores intercambian la tabla de rutas cuando establecen la conexión inicial y periódicamente se envían actualizaciones incrementales de la tabla inicial.

Mensajes BGP

- **OPEN:** Establece la sesión BGP. Incluye identificador de AS y parámetros de configuración.
- **UPDATE:** Actualización incremental de la información de encaminamiento. Cada mensaje puede incluir una red alcanzable en CIDR con sus atributos, incluida la ruta, y una lista de redes retiradas.
- **NOTIFICATION:** Se envía a los vecinos cuando se detecta un error. Implica un cierre de la sesión y la invalidación de las rutas asociadas.
- **KEEPALIVE:** Para asegurar que la sesión permanezca activa. Se envía en respuesta a un mensaje OPEN y periódicamente para informar de la presencia del encaminador. Si pasado un tiempo (hold) no se recibe información, se cierra la sesión.

Bibliografía

- [1] Modelo OSI
https://es.wikipedia.org/wiki/Modelo_OSI
- [2] Protocolos de internet
https://es.wikipedia.org/wiki/Familia_de_protocolos_de_internet
- [3] Direcciones IPv6
<http://ipv4to6.blogspot.com/p/tipos-de-direcciones-ipv6-unicast.html>
- [4] Direcciones IPv6
<http://blog.capacityacademy.com/2013/04/17/cisco-ccna-todo-sobre-ipv6-direcciones/>
- [5] Direcciones IPv6
https://es.wikipedia.org/wiki/Direcci%C3%B3n_IPv6
- [6] NAT
<https://binfalse.de/2011/06/30/connecting-through-a-nat-the-not-trivial-directive/>
- [7] Zonas IPv6
<https://www.ronaldschlager.com/2014/ipv6-addresses-scopes-zones/>
- [8] IPv6
<https://es.wikipedia.org/wiki/IPv6>
- [9] Guía TCP
<http://www.tcpipguide.com/index.htm>
- [10] Vecinos
https://es.wikipedia.org/wiki/Neighbor_Discovery
- [11] Vector de distancias
https://es.wikipedia.org/wiki/Vector_de_distancias
- [12] Estado de Enlace
https://es.wikipedia.org/wiki/Estado_de_enlace
- [13] RIP
https://es.wikipedia.org/wiki/Routing_Information_Protocol
- [14] OSPF
https://es.wikipedia.org/wiki/Open_Shortest_Path_First
- [15] OSPF
<https://networklessons.com/ospf/ospf-hello-and-dead-interval/>

[16] BGP

https://es.wikipedia.org/wiki/Border_Gateway_Protocol

[17] Vector de rutas

https://en.wikipedia.org/wiki/Path_vector_routing_protocol

2019 - Powered by *LATEX*