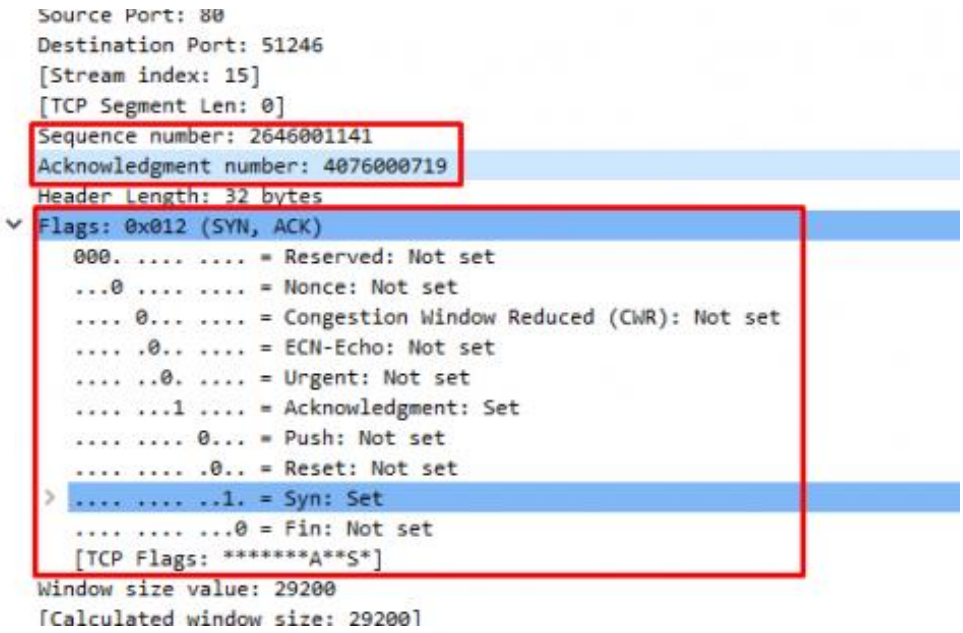
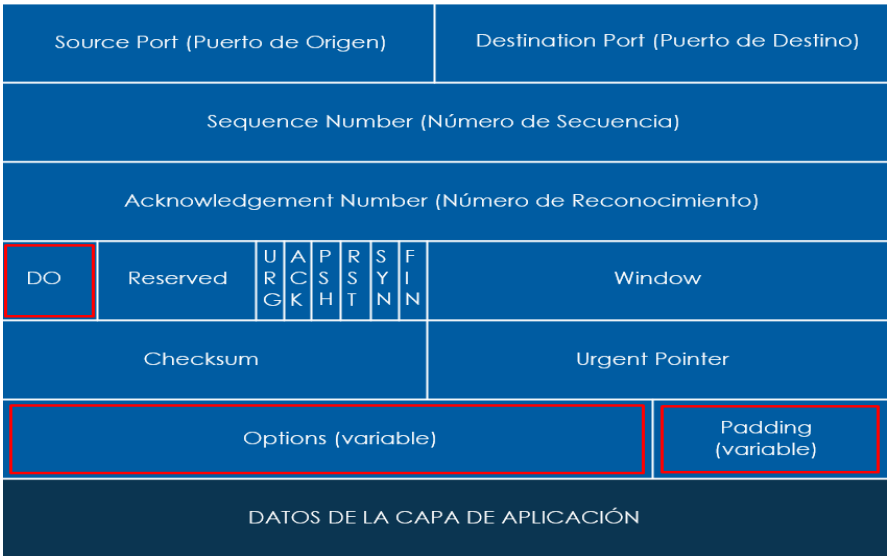
Tarea # 4

Servicios Opcionales (MSS, SACK, …)

En el establecimiento de la conexión TCP, se pueden intercambiar un conjunto de parámetros opcionales para la sesión, muy aparte de los datos de Número de SYN, ACK o Flags.

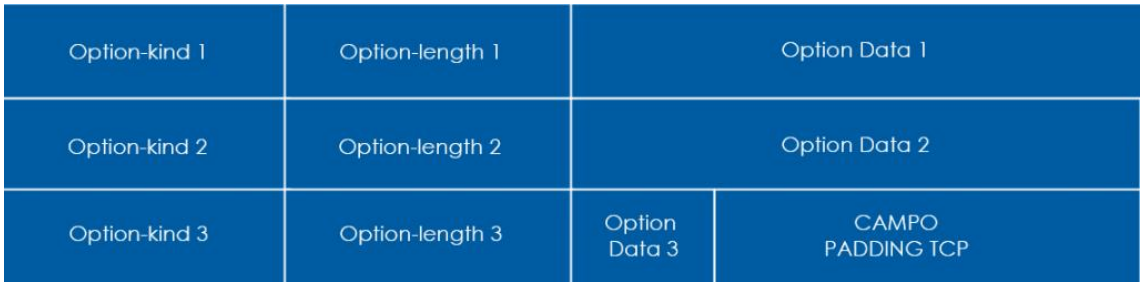


Estos parámetros opcionales ayudan a definir cómo debe operar la sesión TCP. Dichos parámetros se encuentran almacenados en el campo denominado “opciones” del header TCP. El campo «opciones» es de un tamaño de variable permite tener flexibilidad a la hora de añadir diferentes opciones que tienen diferentes propósitos para la sesión TCP. El campo opciones**.**



Estructura campo opciones

Cada opción que se intercambia en el header TCP, comienza con 1 byte que especifica el tipo de opción que se está intercambiando, ese campo se denomina “Option-kind” y continúa con 1 byte que detalla el tamaño del campo opciones en bytes, este campo es denominado “Options-length”. Este campo considera el byte del “option-kind”, el byte de su propio campo “options-length” y los bytes de las opciones en sí que se denominan «Options-Data». Si el tamaño del «Options-length» no corresponde a un múltiplo de 32 bits o 4 bytes, se utiliza el Padding del campo TCP.



La lista de todas las opciones que se pueden intercambiar con TCP te la mostramos en la tabla.

Kind	Length	Meaning	Reference
0	-	End of Option List	[RFC793]
1	-	No-Operation	[RFC793]
2	4	Maximum Segment Size	[RFC793]
3	3	Window Scale	[RFC7323]
4	2	SACK Permitted	[RFC2018]
5	N	SACK	[RFC2018]
6	6	Echo (obsoleted by option 8)	[RFC1072][RFC6247]
7	6	Echo Reply (obsoleted by option 8)	[RFC1072][RFC6247]
8	10	Timestamps	[RFC7323]
9	2	Partial Order Connection Permitted (obsolete)	[RFC1693][RFC6247]
10	3	Partial Order Service Profile (obsolete)	[RFC1693][RFC6247]
11		CC (obsolete)	[RFC1644][RFC6247]
12		CC.NEW (obsolete)	[RFC1644][RFC6247]
13		CC.ECHO (obsolete)	[RFC1644][RFC6247]
14	3	TCP Alternate Checksum Request (obsolete)	[RFC1146][RFC6247]
15	N	TCP Alternate Checksum Data (obsolete)	[RFC1146][RFC6247]
16		Skeeter	[Stev_Knowles]
17		Bubba	[Stev_Knowles]
18	3	Trailer Checksum Option	[Subbu_Subramaniam][Monroe_Bridges]
19	18	MD5 Signature Option (obsoleted by option 29)	[RFC2385]
20		SCPS Capabilities	[Keith_Scott]
21		Selective Negative Acknowledgements	[Keith_Scott]
22		Record Boundaries	[Keith_Scott]
23		Corruption experienced	[Keith_Scott]
24		SNAP	[Vladimir_Sukonnik]
25		Unassigned (released 2000-12-18)	
26		TCP Compression Filter	[Steve_Bellovin]
27	8	Quick-Start Response	[RFC4782]
28	4	User Timeout Option (also, other known unauthorized use) [***][1]	[RFC5482]
29		TCP Authentication Option (TCP-AO)	[RFC5925]
30	N	Multipath TCP (MPTCP)	[RFC6824]
31		Reserved (known unauthorized use without proper IANA assignment) [**]	
32		Reserved (known unauthorized use without proper IANA assignment) [**]	
33		Reserved (known unauthorized use without proper IANA assignment) [**]	
34	variable	TCP Fast Open Cookie	[RFC7413]
35-68		Reserved	
69		Reserved (known unauthorized use without proper IANA assignment) [**]	
70		Reserved (known unauthorized use without proper IANA assignment) [**]	
71-75		Reserved	
76		Reserved (known unauthorized use without proper IANA assignment) [**]	
77		Reserved (known unauthorized use without proper IANA assignment) [**]	
78		Reserved (known unauthorized use without proper IANA assignment) [**]	
79-252		Reserved	
253	N	RFC3692-style Experiment 1 (also improperly used for shipping products) [*]	[RFC4727]
254	N	RFC3692-style Experiment 2 (also improperly used for shipping products) [*]	[RFC4727]

End of Option List (EOL)

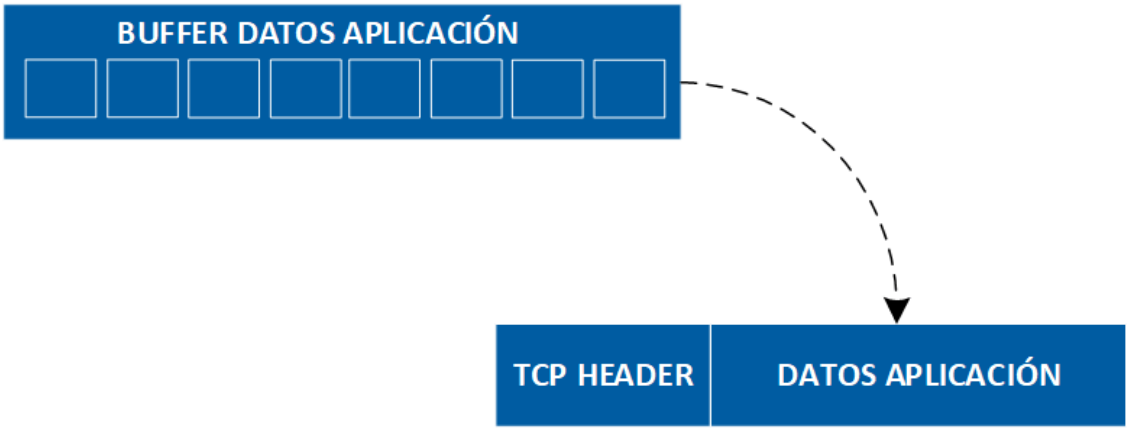
Esta opción no tiene un campo “Options-length”, ni tampoco el campo “Options data”. Se utiliza para hacer conocer que ya no existen más opciones en el header TCP a ser procesadas. Por lo tanto es de un tamaño de 1 byte. Solo se utiliza esta opción si el tamaño del resto de las opciones no termina en un número de byte múltiplo de 4. Si al añadir esta opción (1 byte), no se logra llegar al múltiplo de 4, pues se utiliza el campo Padding del header TCP para llegar al valor requerido.

No-Opration (NOP)

La segunda opción se denomina No-Operation, tiene un campo Options-kind de 1, no utiliza el campo Options-length, ni tampoco lleva bytes de datos. Esta opción se utiliza como un padding opcional para que la subsiguiente opción pueda comenzar en un límite de 4 bytes. Con este campo ya eliminaríamos la necesidad del campo Padding del header TCP, sin embargo no todas las implementaciones de TCP utilizan este campo.

Maximun Segment Size (MSS)

Esta opción se intercambia solamente en el primer Segmento SYN y SYN ACK del establecimiento de la conexión.
Recuerda que cuando un host está construyendo su segmento TCP, agarra los bytes de los datos de aplicación que se encuentran en los buffers y los coloca en la porción de datos del segmento TCP.



La cantidad está definida por el MSS, los 2 dispositivos finales que se comunican tienen que ponerse de acuerdo en el tamaño máximo de los datos del segmento que están dispuestos a recibir. Este MSS se envía en el proceso 3-Way Handshake.

Un tamaño muy grande y muy pequeño de MSS, tiene un impacto directo en el rendimiento de la red. Si se elige un MSS pequeño, la utilización de la red es pobre. Supongamos que elegimos un MSS de 10 bytes, solamente en el header TCP tenemos 20 bytes, por lo tanto, cada segmento lleva 10 bytes de datos y 20 bytes de control. Obviamente no es óptimo.

Por el otro lado que pasa si elegimos un tamaño muy grande de MSS. Un tamaño grande de MSS nos proporciona un segmento grande que obviamente provocara que se tengan paquetes IP grandes. ¿Qué pasa cuando tenemos paquetes IP de gran tamaño? Lo que pasa es que pueden ser fragmentados en el camino al dispositivo de destino debido al MTU.

Control de la Congestión en el Protocolo de Transporte TCP

El control de congestión de TCP es parte fundamental de este protocolo y con los años ha experimentado un proceso de mejora constante a través de la generación de diferentes versiones, como TCP Tahoe, Reno, Vegas, etc.

Es interesante el caso de la versión TCP CUBIC, la cual desde hace algunos años es el control de congestión que aplican por defecto los sistemas Linux/Unix.

TCP CUBIC se hace más interesante aún ya que Microsoft ha decidido que esta versión sea parte fundamental de productos como Windows 10 y Windows Server 2019, tal como se lee en este documento sobre las novedades de Windows Server 2019 y en este sobre Windows 10.

El hecho de tener una misma distribución en los entornos Linux y Windows ha llevado a los administradores de red a repasar la idea tras el control de congestión que plantea TCP y lo que implica TCP CUBIC.

Un punto interesante del control de congestión que aporta TCP es que se trata de procesos con las siguientes características:

- Estos procesos corren solo en los equipos emisores.
- No generan tráfico.
- Propician un reparto equitativo de la capacidad de transmisión de la red. Como cada equipo decide sobre su capacidad de transmisión, atendiendo solo al comportamiento de la red que él observa, no se favorece ni perjudica a ningún equipo emisor bajo ninguna circunstancia.

Ahora bien, es fácil entender que no es justo comparar las capacidades de los algoritmos de control de congestión de TCP con las capacidades de una herramienta de monitorización, porque nos estamos moviendo en dos universos completamente distintos.

Sin embargo, les proponemos repensar el alcance de una herramienta de monitorización de propósitos generales como Pandora FMS desde el ángulo de estos algoritmos.

Así surgen los siguientes puntos:

- El objeto de estudio de una herramienta de monitorización es mucho más amplio: una herramienta de monitorización debe considerar todos los protocolos presentes en la plataforma, no solo TCP.

- La idea de una herramienta de monitorización es incluir bajo su esquema a todos los componentes, ofreciendo siempre una visión global de la plataforma.
- Los mecanismos que utiliza una herramienta de monitorización, tales como los asociados con la administración de red como SNMP o con el control de flujo de tráfico como NetFlow, son protocolos que implican el envío de paquetes asociados con sus funciones. Ahora bien, por supuesto las herramientas de monitorización tienen como objetivo establecer esquema que interfiera lo justo con el rendimiento global de la plataforma.
- La causa raíz de la congestión: el acercamiento que logran las herramientas de monitorización pretende llegar a la causa raíz de la congestión. Quizás la causa de un estado de congestión esté en la configuración errada de un protocolo de enrutamiento, lo que no se va a corregir con que los equipos emisores modifiquen su capacidad de transmisión.
- Para finalizar debemos decir que un objetivo de las herramientas de monitorización es generar información que permita predecir una situación de congestión antes de que aparezca.

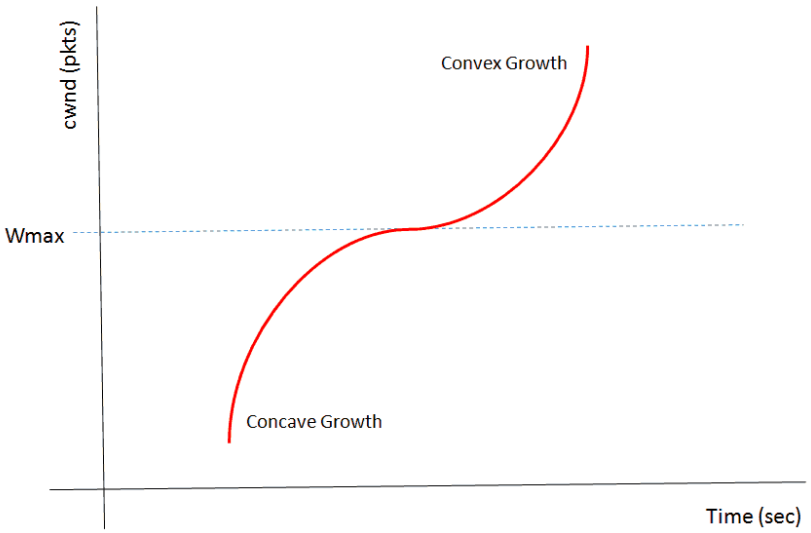
TCP CUBIC

TCP CUBIC es un algoritmo de control de congestión que surge con la idea de tomar ventaja del hecho de que actualmente los enlaces de comunicaciones suelen disponer de niveles de ancho de banda cada vez mayores.

En una red compuesta por enlaces de amplios anchos de banda un algoritmo de control de congestión que lentamente incrementa el ratio de transmisión puede terminar por desperdiciar la capacidad los enlaces.

La intención es disponer de un algoritmo que trabaje con ventanas de congestión cuyos procesos de incremento sean más agresivos, pero que se restrinjan de sobrecargar la red.

Para lograr esto se propone que el esquema de incremento y disminución de la ratio de transmisión se establezca de acuerdo a una función cúbica. Veamos la siguiente figura:



Aplicaciones de Multimedia en Tiempo Real (RTP y VoIP)

La capacidad de las redes en sistemas LAN y WAN ha aumentado considerablemente en estos últimos tiempos. Años atrás, los enlaces no superaban los 2 Mbps, mientras que hoy, las redes operan a 155 Mbps o más.

Esta rápida introducción de redes de alta velocidad ha motivado el desarrollo de nuevas aplicaciones que han sido altamente asimiladas por los usuarios quienes han experimentado cambios, a nivel individual y de empresa, en la forma de operar. La demanda por utilizar estos sistemas de comunicación ha sido también consecuencia del desarrollo de computadores de alta capacidad y de programas con interfaces gráficas adecuadas y simples de usar.

Las características más importantes de una red física son el ancho de banda y el retardo. El ancho de banda queda determinado por la capacidad de los enlaces físicos que se utilicen, mientras que el retardo dependerá de la tecnología utilizada, el largo de los enlaces y el número y la capacidad de los enrutadores que procesen los datos. Sin embargo, dada la estructura con que está diseñada la red Internet, con la congestión aparecen efectos negativos para la transmisión de datos.

Cada enrutador está implementado con un sistema de buffer que permite almacenar y procesar cierto volumen de tráfico. Si la carga aumenta, los buffer se llenan produciendo pérdidas y retraso de paquetes. Dado que el enrutamiento es dinámico, existe la posibilidad que los paquetes tomen caminos diferentes provocando desorden en los datos y jitter , variación en el tiempo de llegada entre paquetes sucesivos.

Es posible clasificar dos tipos de tráfico en la red dependiendo del tipo de aplicación. Se denomina tráfico elástico al tráfico correspondiente a servicios que no se ven muy afectados por las condiciones de la red. Internet fue diseñado para este tipo de tráfico y, en volumen, corresponde a la mayoría de la carga transmitida. Ejemplo de estas aplicaciones son correos electrónicos, transferencia de archivos, etc.

El protocolo de transporte ideal para estas aplicaciones es TCP pues ofrece un servicio confiable orientado a conexión. Por otro lado, se denomina tráfico inelástico al tráfico generado por servicios susceptibles a las condiciones de red como aplicaciones en tiempo real.

Las propiedades de red deseables para este tipo de tráfico son: bajo jitter ; baja latencia; poder integrar tráfico elástico e inelástico; ancho de banda constante; adaptación a los cambios dinámicos de la red; mínima utilización de requerimientos de buffer dentro de la red; baja carga adicional producto de encabezados de protocolos; baja carga computacional en componentes de red, etc. Estos requerimientos son difíciles de entregar en una red como Internet, pues ni TCP ni UDP ofrecen servicios adecuados para este tipo de tráfico.

Protocolo de tiempo real (RTP)

El protocolo de tiempo real (Real Time Protocol , RTP), provee funcionalidades apropiadas para la transmisión en tiempo real de aplicaciones como video y audio sobre servicios entre dos (unicast) o entre varios (multicast) equipos. El protocolo RTP, no provee reserva de recursos ni garantiza QoS, pero permite monitorear la recepción de datos y controlar e identificar el tipo de servicio ofrecido. Su diseño es independiente de los protocolos utilizados en la capa de red.

VoIP

El objetivo es utilizar Internet como una red telefónica, es decir, usar una red de conmutación de paquetes como una red de conmutación de circuitos.

VOZ SOBRE IP (VoIP): Servicio telefónico IP extremo a extremo con teléfonos o terminales IP. Desde el teléfono IP se establece la conexión con el otro teléfono IP (protocolo SIP). Desde el origen, teléfono IP, salen datagramas IP con paquetes o streams RTP (trozos de voz de 20 ms) que se encaminan por Internet o por cualquier red privada IP. Las aplicaciones de telefonía IP más conocidas: Skype, VoIPBuster, Jajah, etc.

TELEFONIA IP (ToIP): Servicio telefónico IP extremo a extremo con teléfonos o terminales “no IP” (teléfonos digitales que emplean tecnología PCM o teléfonos analógicos convencionales) que hacen uso del servicio VoIP mediante “Gateways media” o pasarelas que convierten los paquetes IP en señales digitales o analógicas y viceversa. Desde el origen, teléfono digital, sale una señal PCM (pulsos digitales) hasta el Gateway. Desde el origen, teléfono analógico, sale una señal analógica hasta el Gateway o pasarela.

Podemos observar tres escenarios de VoIP y ToIP:

- Del teléfono IP (o PC) a teléfono IP (o PC): se envían datagramas IP con paquetes RTP de voz (VoIP)

- Del teléfono IP (o PC) a teléfono convencional: se envían datagramas IP con paquetes RTP de voz (VoIP) hasta el Gateway que convierte los datagramas IP en señales analógicas o digitales y viceversa.

- Del teléfono (analógico o digital) a teléfono (analógico o digital): datagramas IP con paquetes RTP de voz (VoIP) entre los gateways de cada teléfono.

Modelos de Calidad de Servicio (Servicios Integrados-RSVP y Servicios Diferenciados)

Con el crecimiento de las aplicaciones multimedia se necesitan intentos serios para garantizar la calidad de servicio o QoS añadiendo un modelo de asignación de recursos en la red.

MODELO DE SERVICIOS INTEGRADOS.

Modelo basado en reservar previamente recursos (caudal y retardo) en la red para cada flujo. Incorpora señalización en redes IP:

- Protocolo RSVP: señalización la reserva de recursos para un determinado flujo por las mejoras rutas.
- Garantías de cumplimiento en cada router para cada flujo en cuanto a caudal y retardo máximo.
- Algunos flujos requieren mas recursos que otros. Se calcula un árbol de enlaces de menor coste (RPM) para cada fuente que cubra a todos los miembros del grupo.

Posteriormente, se informa via protocolo RSVP a cada router implicado en el trayecto. El algoritmo de encaminamiento no es parte de RSVP.

-PROTOCOLO RSVP: Abarca tanto los errores lógicos como físicos. La reserva la hacen los receptores de un flujo, no el emisor. Se basa en que todos los mensajes que pertenecen a un flujo de datos determinado siguen el mismo camino. El receptor necesita saber, previamente, el camino de menor coste para hacer la reserva Las tablas IP configuradas previamente mediante un IGP de multidifusión. Hay dos mensajes básicos en este protocolo.

∗ **PATH:** Va por donde indican las tablas IP previamente configuradas almacenando la dirección del router precedente.

∗ **RESV:** Hace la reserva salto a salto siguiendo la dirección del router precedente indicado en el mensaje PATH (en cada salto, el router anota en su tabla la reserva indicada) La señalización es usuario-red (mensaje RSVP encapsulados en UDP) y router-router (mensajes RSVP encapsulados directamente en IP)

Una comunicación interactiva entre dos terminales puede requerir de reservas diferenciadas en ambas direcciones. Si la comunicación consta de varios flujos, requiere una reserva para cada flujo. Para obtener una mejor recepción, cualquier de los receptores de un grupo puede enviar un mensaje de reserva por el árbol al emisor. El mensaje se propaga por la ruta inversa del árbol de expansión. En cada salto, el router anota en su tabla la reserva indicada (reservando buffers y caudal y si no puede: informa del fallo en el mensaje). Si un router ya ha reservado recursos para un receptor, si recibe un mensaje RSVP de otro receptor del mismo grupo, no necesita reserva otra vez.

MODELO DE SERVICIOS DIFERENCIADOS

A efectos prácticos se usa DiffServ. No requiere una configuración avanzada, ni reserva previa de recurso ni negociación extremo a extremo que consume tiempo para cada flujo.

La calidad de servicio basada en la clase del servicio. Utiliza una codificación, DSCP, de 6 bits que es la misma tanto para IPv4 e IPv6. Es el típico modelo para un grupo de routers que forman un dominio administrativo (ISP): la administración define un conjunto de clases de servicio; los datagramas IP del cliente que entran en un dominio contienen un campo de tipo de servicio solicitando una clase de servicio determinado y previamente contratado. El router encamina por la dirección de destino del paquete, ofreciendo los recursos indicados por la clase de servicio.

-DSCP: En IPv4 son los 6 bits (Tipo de Servicio, TOS) que van a continuación del campo “longitud cabecera”. En IPv6 el campo DSCP ocupa del bit 4 al 10 del primer octeto, una vez ampliado el campo prioridad a 8 bits. El valor DSCP (6 bits), indica el funcionamiento por salto PHB o clase de servicio que se ha de aplicar.