

Arquitecturas de Calidad de Servicio: DiffServ e IntServ

Planificación y gestión de redes de ordenadores

Departamento de Teoría de la Señal y Comunicaciones y
Sistemas Telemáticos y Computación (GSyC)

Septiembre de 2014



Transparencias adaptadas del material de los libros
que aparecen referenciados al final del fichero
©2014 Grupo de Sistemas y Comunicaciones.
Algunos derechos reservados.
Este trabajo se distribuye bajo la licencia
Creative Commons Attribution Share-Alike
disponible en <http://creativecommons.org/licenses/by-sa/3.0/es>

- 1 Introducción
- 2 IETF Differentiated Services
- 3 IETF Integrated services
- 4 DiffServ vs IntServ
- 5 Referencias

Contenidos

- 1 **Introducción**
- 2 IETF Differentiated Services
- 3 IETF Integrated services
- 4 DiffServ vs IntServ
- 5 Referencias

DiffServ vs IntServ

- **DiffServ**: Los paquetes se marcan a la entrada de la red DiffServ, según diferentes categorías o clases. Entre diferentes clases se establecen diferentes parámetros de QoS. En una misma clase se agregan diferentes flujos que recibirán el mismo tratamiento de QoS.
- **IntServ (RSVP)**: antes de utilizar la red es necesario solicitar una reserva de recursos a lo largo de todo el camino por el que circularán los paquetes, desde el origen al destino. Se obtiene una garantía de que se va a conseguir cierta QoS.

Contenidos

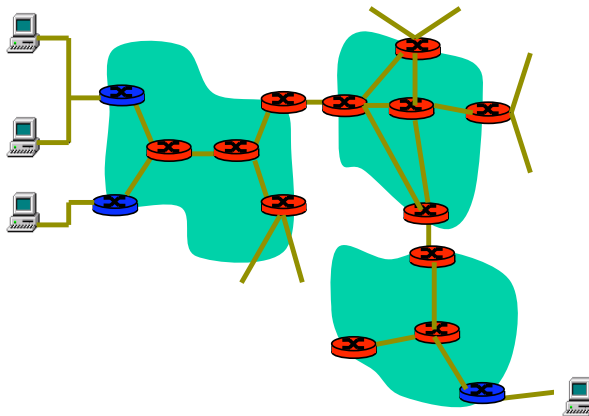
- 1 Introducción
- 2 IETF Differentiated Services**
- 3 IETF Integrated services
- 4 DiffServ vs IntServ
- 5 Referencias

IETF Differentiated Services (DiffServ)

- Buscamos ofrecer clases de servicio cualitativas con un comportamiento particularizado para cada clase, como si cada una tuviese un medio de transmisión en exclusiva
- **Escalabilidad:** funciones simples en el núcleo de la red y relativamente complejas en los **routers de frontera** o en los terminales
 - La **señalización** empleada en redes de conmutación de circuitos como la red telefónica requiere mantener información de estado en el router para cada flujo
 - No escala si hay muchos flujos
- La **arquitectura Differentiated Services (DiffServ) del IETF** no define clases de servicios concretas sino que ofrece componentes funcionales que permiten construir clases de servicio
 - IETF: Internet Engineering Task Force

Arquitectura de DiffServ

- Router frontera
- Router del núcleo de la red

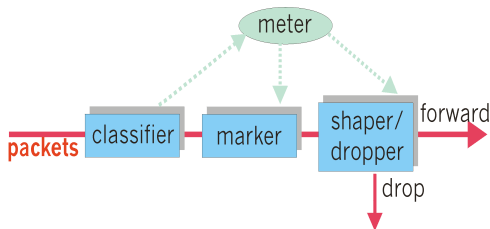


Contenidos

- 1 Introducción
- 2 IETF Differentiated Services**
 - Router Frontera
 - Router del Núcleo
- 3 IETF Integrated services
- 4 DiffServ vs IntServ
- 5 Referencias

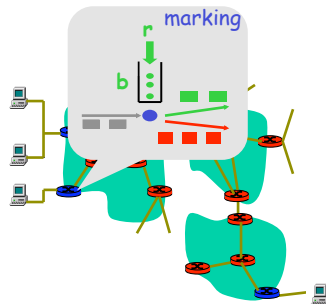
Router frontera (Edge Router)

- Realiza **clasificación** de paquetes y **acondicionamiento** (*shaping/policing*) del tráfico
 - Se usa también un *token bucket* con granularidad de **flujo**
 - En este caso el *token bucket* se usa para adecuar el tráfico inyectado a lo contratado
- **Marca** los paquetes como **in-profile** o **out-profile** en función de si respetan o no lo acordado (límites de tasa promedio, de pico,...)



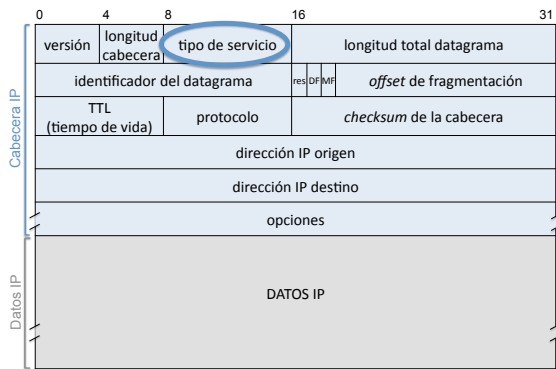
Arquitectura de DiffServ en Router Frontera

- ❶ En primer lugar los paquetes que llegan al *router* de frontera se **clasifican**:
 - El clasificador selecciona los paquetes basándose en los valores de uno o más campos de la cabecera del paquete (dirección origen, destino, puerto origen, destino, identificador de protocolo)
- ❷ A continuación se **marcan** en diferentes categorías o clases.
- ❸ Puede ser deseable limitar la tasa de inyección de tráfico para alguna de las clases de servicio:
 - El usuario **declara un perfil de tráfico** (tasa de pico, tasa promedio, tamaño de ráfaga) usando la abstracción de la cubeta
 - El tráfico es medido y acondicionado (si no respeta el perfil de tráfico declarado se descarta, o se marca como *out-profile*)



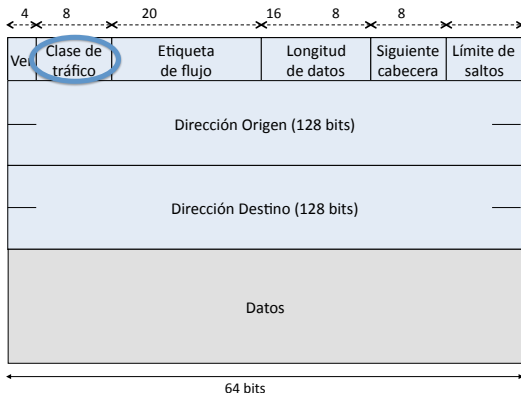
Marcado del tráfico DiffServ en la cabecera IPv4

- Los paquetes se marcan en el campo de 8 bits Type of Service (ToS) de IPv4



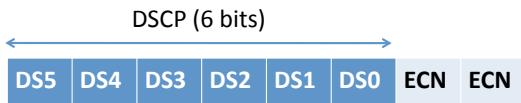
Marcado del tráfico DiffServ en la cabecera IPv6

- Los paquetes se marcan en el campo de 8 bits clase de tráfico de IPv6



Marcado del tráfico DiffServ: campo DSCP

- Se usan 6 bits para identificar *Differentiated Service Code Point (DSCP)* que determinan el comportamiento por salto (PHB, Per-Hop Behavior) que recibirá el paquete en los *routers* de la red DiffServ.
- Quedan los 2 bits menos significativos del campo ToS que no se usan para DiffServ, sino para la notificación de congestión (Explicit Congestion Notification, ECN). ECN es utilizado conjuntamente por los extremos de una conexión TCP y los routers intermedios que usan la disciplina de cola RED, Random Early Detection.



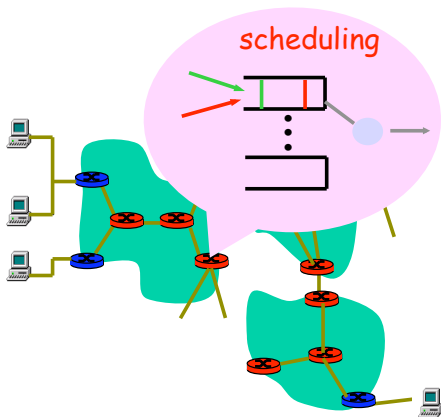
Contenidos

- 1 Introducción
- 2 IETF Differentiated Services
 - Router Frontera
 - Router del Núcleo
- 3 IETF Integrated services
- 4 DiffServ vs IntServ
- 5 Referencias

Router del Núcleo de la red

- Realiza funciones con una granularidad de **clase** de tráfico en función de la marca que trae cada paquete y no en función de cada flujo (origen y destino). Todos los paquetes de cada clase son tratados igual.
- Se encola y se planifica el envío de cada paquete basándose en las **marcas** que pusieron en los paquetes los *routers* de frontera
- Se da preferencia a los paquetes marcados como **in-profile**

Arquitectura de DiffServ en Router del Núcleo de la red



Comportamiento por salto (PHB)

- Cada clase de tráfico (DSCP) está asociado a un comportamiento por salto (*PHB*, *Per Hop Behavior*). El PHB determina el tipo de tratamiento que se le va a dar al paquete en el reenvío:
 - Reserva de recursos: buffer y ancho de banda.
 - Características del tráfico: retardo y pérdidas.
- Existen los siguientes grupos de PHB asociados al campo DSCP:
 - **EF** (Expedited Forwarding): DSCP=101110
 - Bajas pérdidas, baja latencia, bajo jitter, similar a una línea de datos alquilada.
 - **VA** (Voice Admit): DSCP=101100
 - Similar a EF que añade un mecanismo de control de admisión de llamadas.
 - **AF** (Assured Forwarding): 4 clases (AF1, AF2, AF3, AF4)
 - Se proporciona cierta garantía de entrega siempre y cuando se cumpla el acuerdo entre cliente y proveedor sobre el tráfico enviado.
 - Define 4 clases. Prioridad AF4 > Prioridad AF3 > Prioridad AF2 > Prioridad AF1.
 - **DF** (Default Group): DSCP=000000
 - IP Best Effort (compatible con tráfico que no es DiffServ)
 - **CS** (Class Selector): usa los 3 primeros bits DSCP=XXX000 para definir prioridades (compatible con el antiguo ToS).
 - Menor prioridad = 000000 a mayor prioridad = 111000

PHBs definidas actualmente en DiffServ

PHB de Reenvío Acelerado (*Expedited Forwarding (EF)*)

- EF: *“La tasa de salida de una clase de tráfico de un router tiene que ser igual o mayor que una tasa configurada.”*
- Es decir, durante cualquier intervalo de tiempo a la clase de tráfico se le garantiza que recibirá el suficiente ancho de banda como para que la tasa de salida del tráfico sea igual o mayor que la tasa mínima configurada
- Se proporciona la ilusión de un enlace lógico con una tasa mínima garantizada
- El comportamiento por salto EF implica cierta forma de aislamiento entre las clases de tráfico, ya que esta garantía se implementa independientemente de la intensidad de tráfico del resto de clases
- Se puede implementar en los *routers* de diversas formas: dando prioridad estricta a los paquetes EF o dándole peso a una cola EF si se usa weighted fair queuing (WFQ)

PHBs definidas actualmente en DiffServ

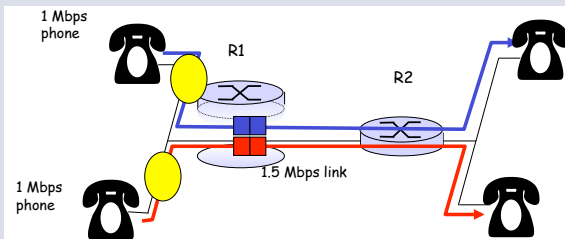
PHB de Reenvío Garantizado (*Assured Forwarding (AF)*)

- AF es más complejo que EF
- Define cuatro clases de tráfico con las que se marcan los paquetes en los *router* frontera usando el DSCP: AF1, AF2, AF3, AF4
- A cada clase le garantiza que recibirá cierta cantidad mínima de ancho de banda y *buffering* usando WFQ
- Variando la cantidad de recursos asignados a cada clase (pesos de colas en WFQ) un ISP puede proporcionar diferentes niveles de rendimiento a las distintas clases de tráfico AF con el siguiente criterio:
 - Prioridad AF4 > Prioridad AF3 > Prioridad AF2 > Prioridad AF1

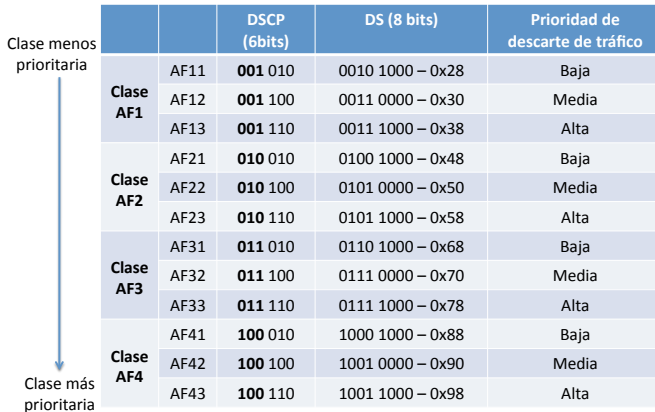
PHBs definidas actualmente en DiffServ

PHB de Reenvío Garantizado (*Assured Forwarding (AF)*)

- Cuando en una clase AF se produce congestión (aumenta el tamaño de las colas), un *router* elimina paquetes basándose en sus valores de preferencia de eliminación
- Dentro de cada clase los paquetes vuelven a clasificarse en tres categorías de **preferencia de eliminación** (o prioridad de descarte), usándose el DSCP para determinar la categoría (4 clases \times 3 categorías de preferencia de eliminación = 12 valores de DSCP distintos).



Clases de tráfico AF



		DSCP (6bits)	DS (8 bits)	Prioridad de descarte de tráfico
Clase AF1	AF11	001 010	0010 1000 – 0x28	Baja
	AF12	001 100	0011 0000 – 0x30	Media
	AF13	001 110	0011 1000 – 0x38	Alta
Clase AF2	AF21	010 010	0100 1000 – 0x48	Baja
	AF22	010 100	0101 0000 – 0x50	Media
	AF23	010 110	0101 1000 – 0x58	Alta
Clase AF3	AF31	011 010	0110 1000 – 0x68	Baja
	AF32	011 100	0111 0000 – 0x70	Media
	AF33	011 110	0111 1000 – 0x78	Alta
Clase AF4	AF41	100 010	1000 1000 – 0x88	Baja
	AF42	100 100	1001 0000 – 0x90	Media
	AF43	100 110	1001 1000 – 0x98	Alta

Cuando se usa el servicio AF, la red DiffServ puede aplicar policing/shaping en la entrada a la red marcando el tráfico del usuario con diferentes prioridades de descarte dentro de una clase, dependiendo de la cantidad de tráfico que no cumpla el contrato establecido.

Contenidos

- 1 Introducción
- 2 IETF Differentiated Services
- 3 IETF Integrated services**
- 4 DiffServ vs IntServ
- 5 Referencias

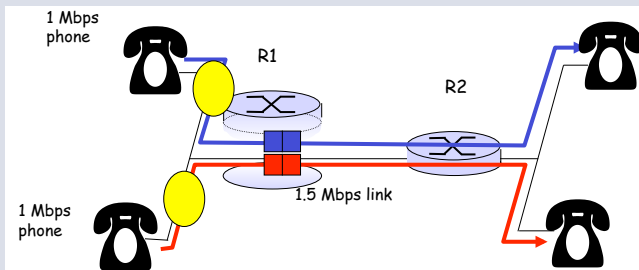
Introducción

- El marcado y vigilancia de paquetes, el aislamiento de tráfico y la planificación en el nivel de enlace pueden proporcionar una clase de servicio con mejor rendimiento que otra
- Con ciertas disciplinas de planificación como la planificación con prioridad, las clases de tráfico inferiores son prácticamente “invisibles” para la clase de tráfico con prioridad más alta
- Con un **dimensionamiento apropiado de la red**, la clase de servicio de más alta prioridad puede conseguir tasas de pérdida de paquetes y retardos extremadamente bajos
- Pero, ¿puede la red garantizar que un flujo activo determinado de alta prioridad continuará recibiendo dicho servicio independientemente de la congestión la red? ¡NO!
- No hay que olvidar que ningún mecanismo puede multiplicar el ancho de banda más allá de la capacidad de cada enlace

Introducción

Principio 4 de la QoS:

Si los recursos necesarios no siempre van a estar disponibles y es necesario garantizar la calidad de servicio, se necesita un proceso de **admisión de llamadas** en el que los flujos declaren sus requisitos de QoS y, o bien se les admita en la red, o se les rechace en función de si la red puede o no proporcionar la QoS requerida



Integrated Services (IntServ)

- Trabajo del IETF entre 1995 y 1997
- Desarrollaron especificaciones para un conjunto de **clases de servicio** diseñadas para satisfacer los requisitos de diferentes clases de aplicaciones.
- Para ofrecer QoS, IntServ se basa en un modelo de reserva de recursos por flujo en todo el trayecto que siguen los paquetes de dicho flujo.
- Un flujo queda identificado por la dirección IP origen, la dirección IP destino, el protocolo de nivel de transporte y opcionalmente el puerto destino.
- La aplicación es responsable de gestionar la reserva de recursos en la red.

IntServ: Clase de servicio *Guaranteed service*

SERVICIO GARANTIZADO

- Para aplicaciones **no elásticas** que requieren que un paquete nunca se descarte y no llegue tarde
- La red debe garantizar que el máximo retardo que cualquier paquete puede experimentar tiene un valor especificado
- La aplicación puede entonces planificar cuándo comenzar a reproducir después de empezar a recibir datos de un flujo multimedia que va siendo almacenado localmente antes de comenzar la reproducción
- Análogo a EF (expedited forwarding) de DiffServ.

IntServ: Clase de servicio *Controlled load*

SERVICIO DE CARGA CONTROLADA

- Para aplicaciones **elásticas** que toleran retardos
- Estas aplicaciones funcionan bien en redes poco congestionadas, reajustando cuánto *buffering* realizan antes de comenzar a reproducir adaptativamente según los retardos observados dinámicamente
- Emula una red poco cargada aunque haya congestión.
¿Cómo?
 - Con WFQ se aísla el tráfico y con control de admisión se limita la cantidad de tráfico de tipo *controlled load*
- Análogo a AF (assured forwarding) de DiffServ.

Mecanismos de IntServ

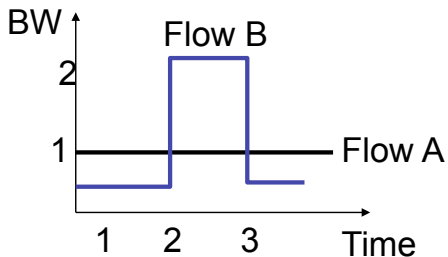
- ❶ **Flowspec:** Se necesita un servicio para comunicar a la red el tipo de servicio requerido por cada aplicación
 - Cualitativamente (ej. “quiero *controlled load*”) o cuantitativamente (“quiero retardo máximo de 100ms”)
- ❷ **Control de admisión:** la red tiene que decidir si puede o no proporcionar el servicio requerido
- ❸ **Protocolo de reserva de recursos (RSVP):** las aplicaciones que generan el tráfico y los elementos de la red tienen que intercambiar información como peticiones de servicio, *flowspecs* y el resultado de las decisiones de control de admisión (llamado señalización en redes de conmutación de circuitos)
- ❹ **Disciplinas de planificación:** los *routers* tienen que cumplir los requisitos especificados en los *flowspec*

- Dos partes diferenciadas:
 - **TSpec (Traffic Specification)**: describe las características del tráfico del flujo que se va a enviar usando la abstracción de la cubeta con fichas (*token bucket*)
 - **RSpec (Request Specification)**: servicio que se requiere de la red. Ej. “*Controlled load service*”. Para “*Guaranteed service*” hay que especificar el límite del retardo.

Ejemplo de *TSpec*

- Un **flujo A** transmite con una tasa constante de 1MBps
 - especificación con la abstracción de la cubeta con fichas:
 $r = 1MBps, b = 1Byte$
- Un **flujo B** transmite con una tasa promedio también de 1MBps pero lo hace transmitiendo:
 - 0'5 MBps durante 2 segundos
 - 2 MBps durante 1 segundo
 - r de la cubeta define la media a largo plazo, luego r también es 1MBps para este flujo, pero para caracterizar el flujo B necesitamos un tamaño b de cubeta de al menos 1MB:
 - durante los 2 primeros segundos transmite a 0'5MBps, ahorrando 1MB, que sumados a 1MB del tamaño de la cubeta le permite enviar 2MBps en el tercer segundo
- Un mismo flujo se puede describir con diferentes combinaciones r, b de cubetas. Ejemplo: el flujo A también se puede describir como el flujo B, pero se trata de usar el que describe de manera más ajustada a la realidad

Ejemplo de TS_{spec}



Flow A: $r = 1$ MBps, $B=1$ byte

Flow B: $r = 1$ MBps, $B=1$ MB

Control de admisión

- Cuando un nuevo flujo quiere recibir un nivel de servicio el control de admisión decide si la TSpec y la RSpec se pueden satisfacer en función de los recursos actuales sin perjudicar al resto de flujos
- La parte difícil es decidir si aceptar o no un flujo.
- Para *guaranteed service* es más difícil salvo que se disponga de WFQ en cada *router*
- Para *controlled load service* se determina en función de heurísticos: “la última vez que permití a un flujo con esta TSpec entrar en esta clase los retardos de la clase de tráfico subieron demasiado así que lo rechazo”

Protocolo de reserva de recursos RSVP

- RSVP: Resource Reservation Protocol
- RSVP es robusto ante fallos de *encaminadores* o enlaces gracias a que usa **soft-state**: la información que almacenan los routers se borra si no se actualiza periódicamente
- Soporta tráfico multidifusión (*multicast*). Origen en aplicaciones de IP multicast como vat y vic
- RSVP es **orientado al receptor**: cada receptor tiene unos requisitos distintos para una misma transmisión
- Soft state + orientado al receptor permite flexibilidad al receptor: es fácil mandar nueva petición con diferente RSpec. La antigua se borra sola pasado un tiempo.

Funcionamiento de RSVP

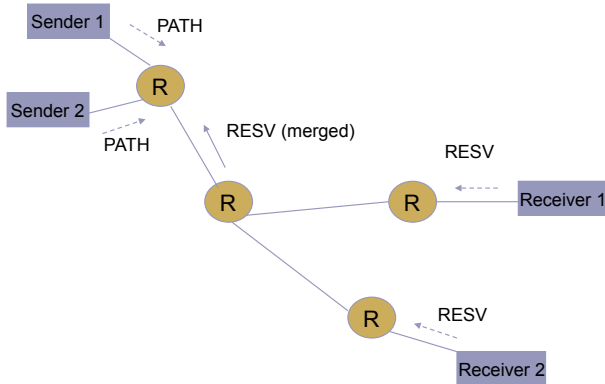
Antes de que el receptor pueda hacer una reserva:

- El receptor tiene que **conocer el tipo de tráfico del emisor** para poder hacer entonces una reserva de recursos adecuada: necesita conocer la TSpec del emisor
- El receptor necesita **conocer la ruta** que recorren los paquetes desde el emisor al receptor para instalar en los *routers* de ese camino las reservas.
 - El camino de ida no tiene por qué ser igual al camino de vuelta.

Funcionamiento de RSVP

- ❶ El emisor manda un mensaje **PATH** a los receptores por IP multicast con la TSpec.
- ❷ Cada encaminador por el que pasa el mensaje PATH se apunta el camino inverso para poder hacer llegar luego mensajes de reserva de los receptores al emisor.
 - RSVP no es un protocolo de encaminamiento, no calcula rutas. Los paquetes siguen el camino que indica la tabla de encaminamiento.
- ❸ Tras recibir el mensaje PATH el receptor envía de vuelta un mensaje **RESV** con la TSpec recibida del emisor y la RSpec con los requisitos del receptor.
- ❹ El paquete **RESV** viaja salto a salto, realizando el camino inverso de PATH. Cada *router* por el que pasa en el camino de vuelta decide si puede satisfacer la RSpec y la propaga corriente arriba hacia el emisor, o la rechaza informando corriente abajo al receptor rechazado
- ❺ El receptor tiene que actualizar periódicamente cada 30 segundos la reserva con mensajes RESV

Funcionamiento de RSVP



RSVP: comportamiento con varios receptores

- Aprovecha el encaminamiento IP multicast
- Múltiples receptores, un único emisor: se fusionan corriente arriba las reservas de varios receptores corriente abajo

- Ejemplo:

- El receptor A necesita reserva con retardo $< 100\text{ms}$
 - El receptor B necesita reserva con retardo $< 200\text{ms}$

Si un router recibe primero la reserva de A, si puede admitirla, la propaga corriente arriba hasta llegar al emisor. Si a continuación recibe la reserva de B, no hace falta que el *router* propague corriente arriba esta segunda reserva pues está incluida en la propagación que hizo de la reserva de A.

- Ejemplo:

- El receptor A necesita reserva con retardo $< 100\text{ms}$
 - El receptor B necesita reserva con retardo $< 50\text{ms}$

Si un router recibe primero la reserva de A, si puede admitirla, la propaga corriente arriba hasta llegar al emisor. Si a continuación recibe la reserva de B, comprueba si puede admitido y en caso afirmativo propaga dicha reserva hacia el emisor.

RSVP: comportamiento con varios emisores

- Múltiples emisores, múltiples receptores:
 - Los receptores tienen que recoger las TSpecs de todos los emisores y realizar una reserva suficientemente grande para todos los emisores
 - Depende de la aplicación: en una videoconferencia con 10 participantes podemos suponer que sólo 2 hablan a la vez \Rightarrow receptores reservan para dos emisores y no para 10

RSVP: comportamiento ante fallos de *routers*

- Los protocolos de encaminamiento descubrirán **nuevas rutas del emisor al receptor**
- Los mensajes PATH se envían también cada 30 segundos, pero un *router* los envía antes si descubre que hay cambios en su tabla de encaminamiento
- En cuanto el receptor reciba el nuevo PATH podrá utilizar el nuevo camino de vuelta corriente arriba para mandar sus RESV, lo que reestablecerá las reservas, caducando automáticamente las del camino que se abandona
- Por tanto RSVP se adapta bien a los cambios en la topología

Contenidos

- 1 Introducción
- 2 IETF Differentiated Services
- 3 IETF Integrated services
- 4 DiffServ vs IntServ**
- 5 Referencias

DiffServ vs IntServ

- Aislamiento del tráfico:
 - DiffServ: por clase de tráfico, agregado de varios flujos.
 - IntServ: por flujo.
- Ámbito de QoS:
 - DiffServ: dentro del dominio
 - IntServ: entre origen y destino
- Complejidad en la configuración:
 - DiffServ: configuración realizada a largo plazo para cada categoría, de forma estática.
 - IntServ: configuración realizada por flujo, en el momento en el que se necesita, de forma dinámica. Existen mensajes de señalización entre los routers.
- Escalabilidad:
 - DiffServ: en los routers frontera se mantiene información para cada flujo o agregados de flujos, en los routers del núcleo se mantiene información por cada clase.
 - IntServ: cada router mantiene información de estado por cada flujo.

Contenidos

- 1 Introducción
- 2 IETF Differentiated Services
- 3 IETF Integrated services
- 4 DiffServ vs IntServ
- 5 Referencias**

Referencias

- James F. Kurose y Keith W. Ross, **Redes de Computadores: un enfoque descendente**, Pearson Educación, 5ª edición.
- Larry L. Peterson, Bruce S. Davie, **Computer networks, a systems approach**, edition 4. Morgan Kaufmann 2007.