

Resuelto

PD1(10 p) IPv6. La idea es dar una pequeña discusión comparativa entre las cabeceras de IPv4 e IPv6, incluso más breve de lo que aquí se muestra pero sí al menos comparando y explicando algún campo concreto. Tres ideas de acuerdo a lo que se pide: en qué consiste, transformaciones y limitaciones.

1º Consiste en hacer ambas versiones de IP compatibles transformando (esto es traduciendo) los campos de una en otra cuando nos encontramos con un router que solo habla una de ellas. 2º Resumir estos cambios: En el campo versión se modifica el 4 por el 6 (o viceversa según interese), Tipo de servicio y clase pueden sustituirse, el campo etiqueta de flujo e identificación/flags/posición no son equivalente entre ellos. Es importante la longitud: en el caso IPv6 se debe tomar la longitud total de IPv4 y restarle su cabecera (IHL) pues IPv6 tiene cabeceras de tamaño fijo, protocolo y cabecera siguiente son prácticamente equivalentes como también lo son el TTL y límite de saltos. Las direcciones IPv4 son más cortas luego la transformación a IPv6 es más fácil que a la inversa por ejemplo rellenando con ceros, en cualquier caso es posible con mecanismo externos similares a DNS. 3º Limitaciones: Hay campos de los que hemos hablado de tamaño diferentes así como los tipos de opciones, de modo que se perderá información que potencialmente pueden ser de interés/necesarias, tampoco la relación de las direcciones IP son triviales, y habría que lidiar con el problema de los fragmentos IPv4.

C1 (5 p) Equidad TCP.

La ventaja para tal usuario es que la equidad TCP se da por conexión no por usuario, luego la tasa agregada alcanzada será proporcional al número de conexiones, y por tanto mayor.

Las desventajas son menos claras, quizá el sobrecoste que conlleva mantener varios sockets y conexiones abiertas. Aunque nuestro temario incluye los niveles 3 y 4, es cierto que el uso de varias conexiones TCP obliga a alguna gestión del fichero como partir en *chunks* que puede conllevar algún trabajo a nivel de aplicación al tener que juntarlos, a nivel 3 o 4 no.

Las ventajas para la red son difíciles de encontrar, de hecho podemos considerar que ninguna. Las desventajas es que puede haber usuarios de la red que reciban menos de los que les correspondería si nadie usara múltiples conexiones TCP paralelas.

C2 (10 p) Algoritmos de enrutamiento intra-AS. Hay multitud de soluciones sin *pillarnos los dedos* con los casos frontera, hacemos $L_{ac}=1$, y que, por ejemplo, $L_{be}=3$ (por el router E); ponemos $L_{df}=1$ (para forzar la ruta por i4 para D), y de manera similar (forzar que a G se llegue por 1), fijamos $L_{dg}=3$.

P1 (30p) Tablas de reenvío IP.

A) Dada la siguiente tabla dar una equivalente con 3 entradas:

Direcciones	Interface
188.14.27.64/26	1
188.14.27.128/26	1
188.14.27.192/26	1
0.0.0.0/0	2

Tabla reenvío P1.A

Dos soluciones intuitivas, primero juntar las entradas 2 y 3 en un bloque de 128 direcciones, o bien asignar todo el último octeto a la interface 1 y quitarlo que hemos añadido de más, esto es:

Direcciones	Interface
188.14.27.64/26	1
188.14.27.128/25	1
0.0.0.0/0	2

6

Direcciones	Interface
188.14.27.0/24	1
188.14.27.0/26	2
0.0.0.0/0	2

B) Dada la siguiente tabla dar una equivalente con 2 entradas:

Direcciones	Interface
188.14.27.0/26	1
188.14.27.64/26	1
188.14.27.128/26	1
188.14.27.192/26	1
188.14.26.0/24	1
0.0.0.0/0	2

Tabla reenvío P1.B

Las 4 primeras entradas representan 64*4 direcciones, y la quinta otras 256, podemos juntarlas en un único rango de 512 así:

Direcciones	Interface
188.14.26.0/23	1
0.0.0.0/0	2

C) [...] la Interface 3 tenga exactamente 8 direcciones IP, y Interface 4 haya exactamente 48 direcciones IP. Facilite esta nueva tabla de reenvío sabiendo que no puede tener más de 5 entradas. Explique cada una de estas nuevas entradas.

Direcciones	Interface
188.14.33.0/24	1
0.0.0.0/0	2

Tabla reenvío P1.C

Las soluciones son muchas siendo los dos casos más intuitivos:

- 1) Asignar un rango de 64 direcciones dentro del rango de la interface 1 a la interface 4 y devolver 16 a la interface 1 y luego coger 8 para la interface 3 (pueden ser de las 16 que pretendíamos devolver a la interface 1). Esto es:

Direcciones	Interface
188.14.33.0/24	1
0.0.0.0/0	2
188.14.33.0/26	4 // cojo 64 direcciones
188.14.33.0/28	1 // devuelvo 16
188.14.33.0/29	3 // cojo 8 direcciones sobre esas 16 o no

- 2) Asignar un rango de 32 y otro de 16 disjuntos para la interface 4, y un tercer rango distinto de 8 direcciones para la interface 3:

Direcciones	Interface
188.14.33.0/24	1
0.0.0.0/0	2
188.14.33.0/27	4 // cojo 32 direcciones
188.14.33.32/28	4 // cojo 16 direcciones
188.14.33.48/29	3 // cojo 8 direcciones

P2 (20p) Enrutamiento en Internet.

No se nos da información de reglas impuestas por el gestor de red así que tenemos de acuerdo a BGP:

0) Gestor de red (no aplica)

1) Menor número de ASs intermedios (inter-AS)

2) Menor coste intra-AS (de acuerdo a RIP o OSPF según el caso)

3) Otras: Desconocidas

4) Azar

A) C→E, tanto I1 como I2 son respuestas válidas: 1) nos dice que vayamos por AS4, aplicando 2) vemos para ir a ese router de borde las dos interfaces dan el mismo coste (2 saltos), así que ambas válidas con los datos disponibles (de forma práctica se puede indicar que se escogería al azar, la primera que haya llegado, mayor número de identificación, etc.).

B) A→E, tanto I1 como I2 son respuestas válidas: 1) nos dice que vayamos por AS4, aplicando 2) para ir a ese router de borde las dos interfaces dan el mismo coste (10, cualquiera que sea la métrica definida para OSPF ejecutando Dijkstra), así que valen ambas (o se ejecuta al azar, la primera que haya llegado, mayor número de identificación, etc.).

C) B→D, I1: 1) nos dice que la ruta por AS1 y AS3 implican el mismo número de ASs intermedios, aplicando 2) vemos que RIP (que asigna coste 1 a todos los enlaces) habrá determinado que para ir al router de borde que está arriba se va por la interface 1 en un salto, y al otro router de borde en dos saltos por la interface 3. Se obtiene un menor coste intra-AS escogiendo el router de arriba, esto es, asociando la interface 1 al destino D.

D) A→C, I2: 1) nos dice que la ruta por AS2 y AS4 implican el mismo número de ASs intermedios, aplicamos 2) y resulta que OSPF habrá determinado que para ir al router de borde que está a la izquierda se va por la interface 2 con coste 5, frente a coste 6 que daría lugar el otro router de borde por la interface 1, escoge por tanto por menor coste intra-AS la interface 2.

P3 (25p) TCP y ventana de congestión.

A) En general la tasa se calcula como: $\text{Tamaño_medio_ventana} * \text{Tamaño_unidad_datos} / \text{RTT}$, concuidado de no mezclar unidades y responder en Mb/s.

Aunque en el caso TCP ya sabíamos que su tasa es igual a $\frac{3}{4} * 20$ (Tamaño_máximo_ventana) * MSS * 8 (bits/Bytes) / RTT → 7.8 Mb/s.

Experimento1: $17.16 * 65000 * 8 / 10^6 \rightarrow 8.9 \text{ Mb/s}$

Siendo el tamaño_medio_ventana = $(10+14+17+18+19+19.5+19.75+20)/8$

Experimento2: $13.46 * 65000 * 8 / 10^6 \rightarrow 7 \text{ Mb/s}$

Tamaño_medio_ventana = $(10+14+17+18+19+19.5+19.75+20)/8$

Experimento3: Igual que TCP.

- B)** En la transferencia de un fichero de gran tamaño que implica un camino multi-salto en el que se ignoran las capacidades de los enlaces implicados, TCP intentará transmitir a la máxima velocidad (equivalente a tener la máxima utilización) probando a aumentar la velocidad hasta que haya indicios de pérdida de paquetes, cuando se den, se interpretará como una señal de que se está llegando a la congestión (típicamente 3 ACKs duplicados), se baja la tasa y vuelta a buscar el límite.

Es por ello que cuanto más cerca estemos de este umbral, mejor, ese es el ideal, aunque de forma realista y desconociendo el resto el tráfico de la red, este umbral simplemente se estima *probando y probando*.

En el ejemplo que se nos presenta vemos que este límite parece constante e igual a unos 10.4 Mb/s, ese es el objetivo a conseguir (aunque en escenarios reales este ideal no será un número constante, sino que variará con el tiempo según el resto de usuarios, el tipo de tráfico (CBR, o variable), etc).

Este límite o “embudo” del camino es consecuencia de un enlace de poca capacidad o el uso de otros usuarios del mismo, pero no queremos que el nivel 4 rebaje tal límite “artificialmente” más.

El protocolo que muestra mayor tasa es el del experimento 1, por ello será nuestra elección.

Es cierto que genera alguna pérdida más y quizá ante escenarios en los que la latencia o el *jitter* sean importantes puede representar un inconveniente, no es nuestro escenario pero si merece ser comentado, también quizá hay más trabajo en el extremo receptor al haber más desorden.

De hecho el protocolo experimento 1 parece diseñado a medida a la red que estábamos midiendo y sus condiciones, quizá demasiado y sea ese su mayor inconveniente. Muestra una forma pseudo-logarítmica con tendencia a estabilizar sobre el 20... ¿Qué pasaría si la red cambia y se duplica la capacidad del enlace de menor tamaño, o se descongestiona un enlace saturado en la ruta? Este protocolo muestra poca adaptabilidad para crecer, se quedaría estancado en torno a esos 20 segmentos aunque nos encontráramos ante mejores condiciones. No parece buena idea funciones que tiendan a ser *planas* en escenario cambiantes.

Como comentario final, en esta cuestión se ha valorado suficiente con que se haya facilitado una discusión coherente esto es, que se escoja el protocolo de mayor tasa y se diga que: o bien no hay inconvenientes de importancia, o estos tienen que ver con el mayor número de pérdidas con el impacto que sea mientras tenga sentido, o que el protocolo es poco flexible ante mejoras en las condiciones.