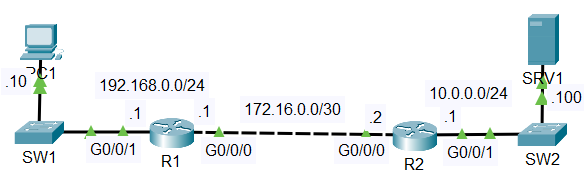
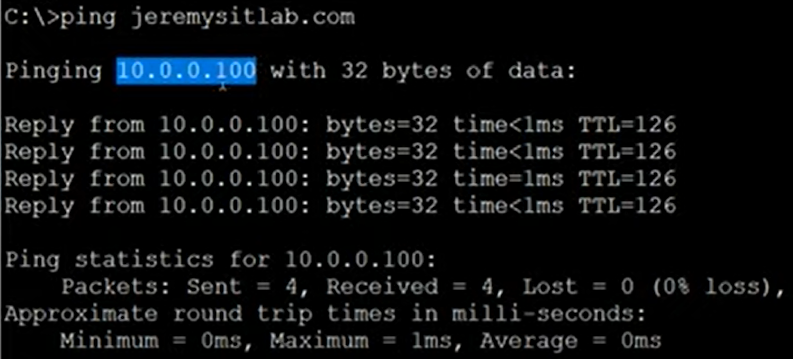
**Quality of service (QoS)**

**Topología**

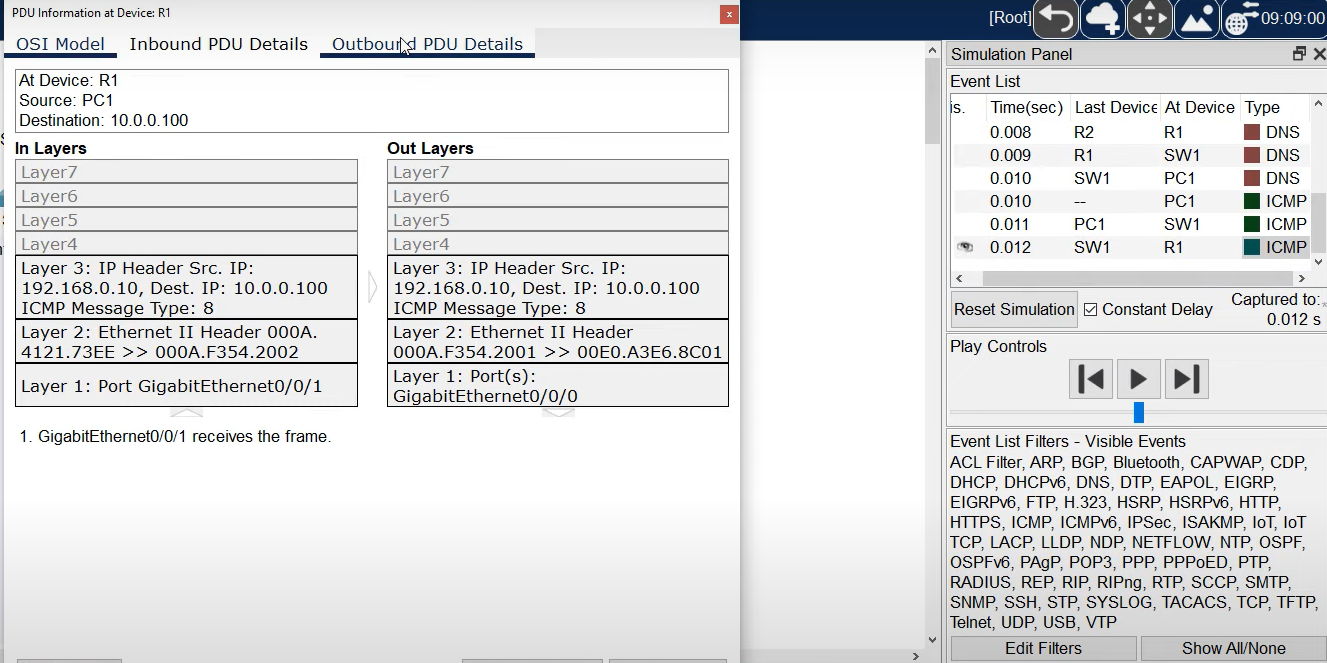
**In **

**Introducción**

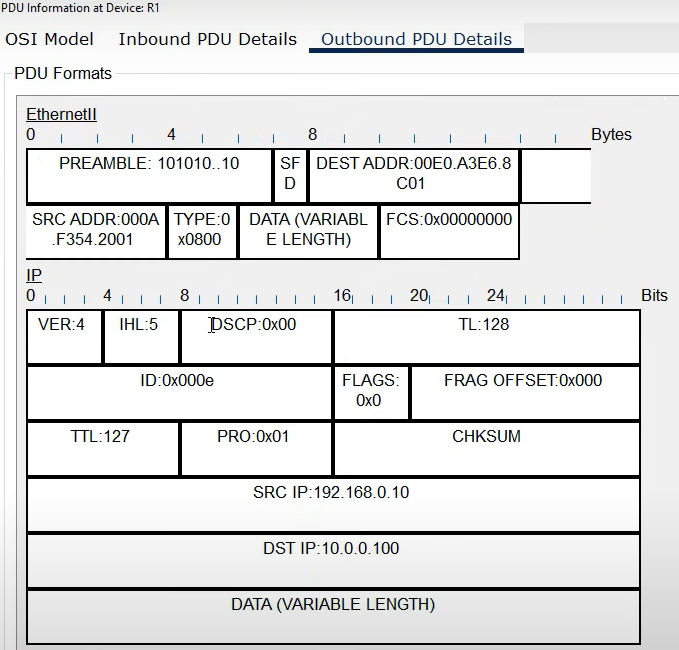
* En este ejercicio, realizaremos algunas configuraciones básicas de QoS en R1. En esta red tenemos la PC1 conectada a R1 a través del switch SW1. Por otro lado, está el servidor SRV1 conectado a R2 vía SW2.
* Aunque solo he mostrado 1 PC, digamos que hay muchas más conectadas a R1 y la red se está congestionando.
* Configuraremos QoS en R1 para garantizar que el tráfico más importante reciba un tratamiento de alta prioridad.
* Marcaremos el tráfico HTTPS como AF31 y nos aseguraremos de que tenga un ancho de banda mínimo del 10% en una cola de prioridad.
* Marcaremos el tráfico HTTP como AF32 y nos aseguraremos de que obtenga un ancho de banda mínimo del 10 %, pero no estará en una cola de prioridad.
* Marcaremos el tráfico ICMP como CS2 y nos aseguraremos de que obtenga un ancho de banda mínimo del 5%.
* El diseño real de QoS de una red depende completamente de la red. Qué tráfico debe priorizarse, cuánto ancho de banda necesita, etc., varía mucho.
* Los valores seleccionados en esta práctica son valores aleatorios que se usan solamente para mostrar la configuración básica de los comandos utilizados en QoS. Dar prioridad al tráfico HTTPS, por ejemplo, probablemente no sea tan común. Por lo general, las colas de prioridad se utilizan para el tráfico de voz.
* Después de configurar las reglas de QoS, las aplicaremos al tráfico que se reenvía desde la interfaz G0/0/0 de R1.
* Cuando R1 reenvía el tráfico de G0/0/0 a R2, se priorizará de acuerdo con nuestras configuraciones.
* Una vez que el tráfico llega a R2, esa es una historia diferente. Incluso si R1 marca los paquetes como de alta prioridad, a menos que R2 también esté configurado para tratar esos paquetes como de alta prioridad, los tratará a todos por igual.
* Configuramos QoS en R1 y priorizará ciertos tipos de tráfico sobre el salto a R2, pero luego, cómo R2 prioriza el tráfico sobre el próximo salto depende de la configuración de R2. No puede simplemente configurar un ruteador para priorizar el tráfico de voz y luego esperar a que el resto de la red también lo priorice. Debe configurar QoS en toda la red, donde sea necesario.
* Para mostrar las configuraciones básicas de QoS, configuraremos R1.
* Antes de la configuración, se mostrarán las marcas predeterminadas de tráfico que se reenvía desde R1.
* Iré a la PC1. Haremos un ping al sitio [www.google.com](http://www.google.com). PC1 enviará una consulta DNS a su servidor DNS, que es SRV1 en 10.0.0.100, y luego aprenderá la dirección IP de jeremysitlab.com, que en este caso también es 10.0.0.100 y luego PC1 hará ping a SRV1. Como puede ver, los pings funcionan.



* Ahora cambiemos al modo de simulación y hagamos ping de nuevo. En realidad, PC1 debería almacenar la dirección IP de jeremysite.com en su caché de DNS, pero parece que no funciona en el packet tracer, por lo que envía la consulta de DNS nuevamente. Vayamos al punto donde está el paquete ICMP en R1. Verifiquemos ese paquete y hagamos clic en los detalles de la PDU de salida.



* Observe que el campo DSCP está escrito en hexadecimal 00. Técnicamente, dos dígitos hexadecimales son 8 bits, pero en realidad el campo DSCP tiene solo 6 bits. Tenga en cuenta que el campo DSCP no está marcado, es 0, reenvío predeterminado.
* Así es como las PC envían su tráfico de forma predeterminada, y los conmutadores y enrutadores no agregarán ninguna marca de forma predeterminada.
* Por cuestiones de tiempo, no le mostraré HTTP y HTTPS ahora, pero también tendrían un valor DSCP de 0.



## Parte 1: Configure los mapas de clase para cada tipo de tráfico

1. Vamos a configurar QoS en R1.
2. Hay tres pasos principales para configurar QoS.
3. **Identificar los tipos de tráfico a los que se desea aplicar un tratamiento especial:** Para dar un tratamiento especial a ciertos tipos de tráfico, debe decirle al ruteador a qué tipo de tráfico desea darle ese tratamiento especial. Para identificar el tráfico en una configuración de QoS de Cisco IOS, usamos mapas de clase. El comando es **class-map**.

**Paso 1: Configura el mapa de clase para el tráfico HTTPS**

* Define a este primer mapa de clase **HTTPS\_MAP**, ya que lo usaremos para hacer coincidir el tráfico HTTPS.

**class-map HTTPS\_MAP**

* Ingrese el comando **match protocol https**. Este mapa de clase se puede usar para identificar el tráfico HTTPS.

**match protocol https**

**Paso 2: Configura el mapa de clase para el tráfico HTTP**

* Define a este primer mapa de clase **HTTP\_MAP**, ya que lo usaremos para hacer coincidir el tráfico HTTP.

**class-map HTTP\_MAP**

* Ingrese el comando **match protocol http**. Este mapa de clase se puede usar para identificar el tráfico HTTP.

**match protocol http**

**Paso 3: Configura el mapa de clase para el tráfico ICMP**

* Define a este primer mapa de clase **ICMP\_MAP**, ya que lo usaremos para hacer coincidir el tráfico HTTP.

**class-map ICMP\_MAP**

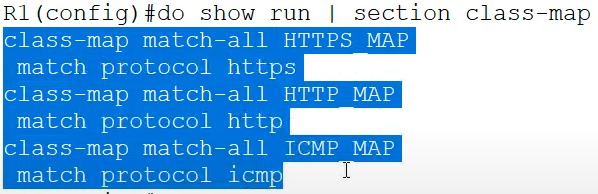
* Ingrese el comando **match protocol icmp**. Este mapa de clase se puede usar para identificar el tráfico ICMP.

**match protocol icmp**

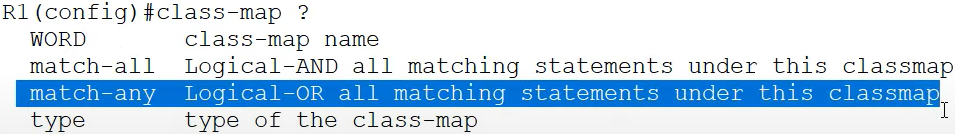
**Paso 4: Verifica la configuración de los mapas de clase**

* Ejecuta el comando **sh run | section class-map.** Se encuentran los tres mapas de clase que se configuraron. En todos los mapas de clase, se aplicó el modo default **math-all.** Esto significa que, para que coincida con este mapa de clase, el tráfico debe coincidir con todas las declaraciones de coincidencia en ese mapa de clase. En este caso, cada mapa de clase solo tiene una declaración MATCH, por lo que no importa, pero si hay varias declaraciones de coincidencia,

Abrir la ventana de configuración



* + - * También puede usar el modo MATCH-ANY, lo que significa que el tráfico coincidirá con ese mapa de clase incluso si coincide solo con uno. de las declaraciones de coincidencia en el mapa de clase. Entonces, podría, por ejemplo, hacer un mapa de clase con dos declaraciones de coincidencia, el match protocol https y match protocol http. En ese caso, si un paquete es un paquete HTTPS o un paquete HTTP, coincidiría con ese mapa de clase.



## Parte 2: Especifique qué tipo de tratamiento queremos dar a cada tipo de tráfico

Hay tres pasos principales para configurar QoS.

1. **Especificar qué tipo de tratamiento queremos dar a cada tipo de tráfico.** Para ello utilizamos mapas de políticas.

**Paso 1: Configura el mapa de políticas para el tráfico HTTPS**

* + - * El comando es **policy-map** y lo nombraremos **g0/0/0\_OUT**, ya que después aplicaremos este mapa de políticas de salida en la interfaz g0/0/0.

**policy-map G0/0/0\_OUT**

* + - * Luego use class, seguido del nombre de uno de los mapas de clase que acabo de configurar, HTTPS\_MAP.
      * Use class, seguido del nombre del mapa de clase: HTTPS\_MAP.

**class HTTPS\_MAP**

* Ahora podemos decirle a R1 qué hacer con el tráfico que coincide con este mapa de clases. Marca cualquier paquete HTTPS con un valor DSCP de AF31

**set ip dscp AF31**

* Luego, le daremos a HTTPS una cola de prioridad con al menos el 10% del ancho de banda de los enlaces durante los períodos de congestión. Esto es todo para la clase HTTPS.

**priority percent 10**

**Paso 2: Configura el mapa de políticas para el tráfico HTTP**

* + - * Use class, seguido del nombre del mapa de clase: HTTP\_MAP.

**class HTTP\_MAP**

* Marca cualquier paquete HTTP con un valor DSCP de AF32

**set ip dscp AF32**

* Ahora le daremos un ancho de banda mínimo del 10% en momentos de congestión, pero no será una cola de prioridad.

**bandwidth percent 10**

**Paso 3: Configura el mapa de políticas para el tráfico ICMP**

* + - * Use class, seguido del nombre del mapa de clase: ICMP\_MAP.

**class ICMP\_MAP**

* Marca cualquier paquete ICMP con un valor DSCP de cs22

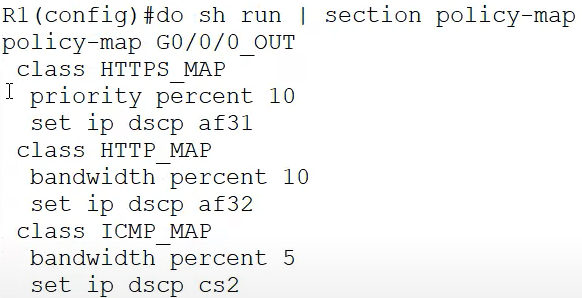
**set ip dscp cs2**

* Ahora le daremos un ancho de banda mínimo del 5% en momentos de congestión y no está en una cola de prioridad.

**bandwidth percent 5**

**Paso 4: Verifica la configuración de los mapas de políticas**

* Ejecuta el comando **sh run | section class-map.** Todavía no se ha aplicado el mapa de política en la interfaz de salida en G0/0/0, pero después de hacerlo, verificará todo el tráfico que se reenviará desde G0/0/0.
* Si es un paquete HTTPS, lo marcará como DSCP AF31 y le dará una cola de prioridad con al menos el 10% del ancho de banda del enlace.
* Si es un paquete HTTP, lo marcará como DSCP AF32 y le dará una cola con al menos el 10% del ancho de banda del enlace.
* Si es un tráfico ICMP, lo marcará como DSCP CS2 y le dará una cola con al menos el 5% del ancho de banda del enlace.
* El resto del tráfico, que no coincida con ninguno de estas políticas, no se marcará y se reenviará sin ningún tratamiento especial de QoS.



## Parte 3: Aplique este mapa de políticas usando una política de servicio

Hay tres pasos principales para configurar QoS.

1. **Aplicar el mapa de políticas usando una política de servicio**

**Paso 1: Aplique el mapa de políticas a la interface g0/0/0 de salida**

**int g0/0/0**

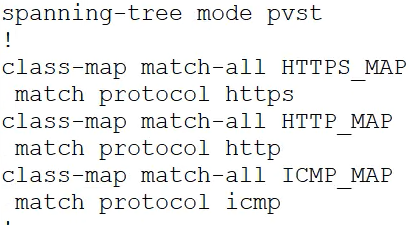
**service-policy out**

**service-policy output g0/0/0\_OUT**

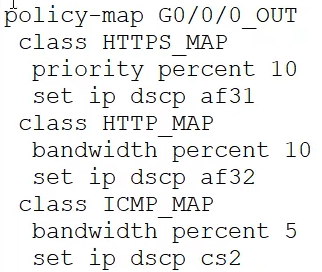
Nuestra configuración QoS está completa.

**Paso 2: Verifica la configuración**

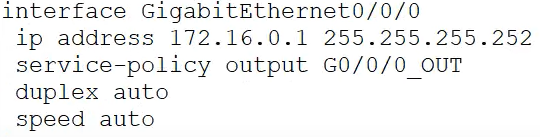
* Ejecuta el comando **sh run.**
* Revisa los **mapas de clase**, identificando los tipos de tráfico que queremos hacer coincidir.



* Debajo de ellos está el **mapa de políticas**, que establece la acción que queremos tomar sobre el tráfico coincidente en los mapas de clase.



* Finalmente, se utilizó una **política de servicio** para aplicar el mapa de políticas en la interface G0/0/0 de salida.



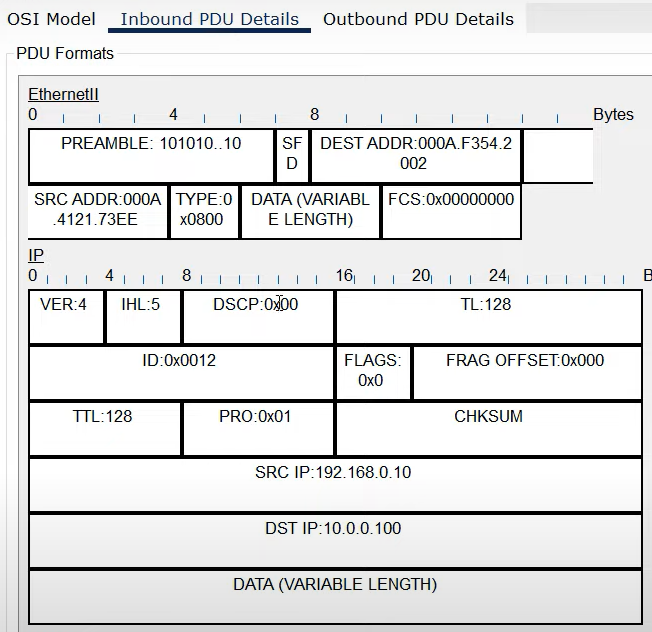
## Parte 4: Envíe tráfico ICMP de la PC1 al servidor SRV1 y verifique las marcas.

**Paso 1: Abra el modo de simulación y haga un ping al servidor SRV1.**

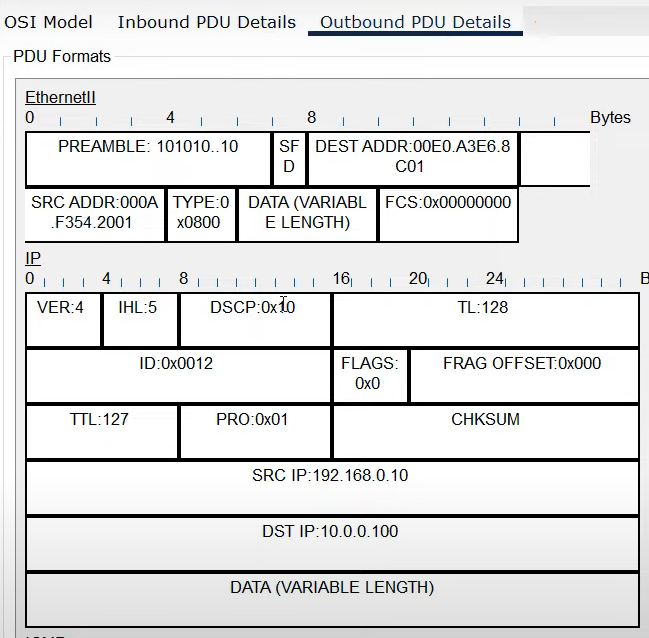
**ping 10.0.0.100**

Llevemos el paquete a R1 y luego verifiquemos.

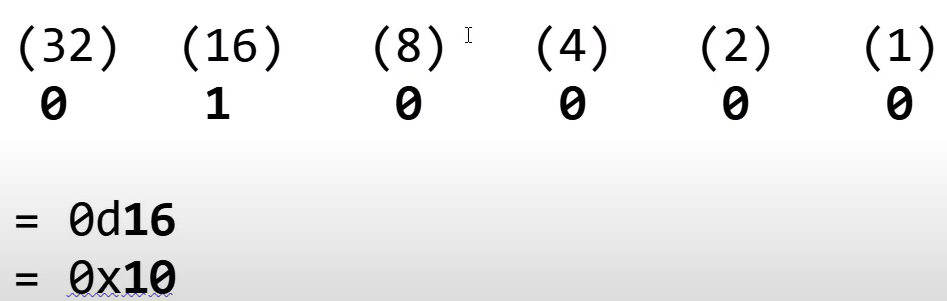
* En la pestaña de **inbound PDU details**, la marca de DSCP es 0.



* En la pestaña de **outbound PDU details**, la marca de DSCP es 1 0. Pero si verificamos la salida, que es el estado del paquete tal como lo reenvía R1, la marca ahora es hexadecimal 1 0.



* Entonces, lo marcamos como CS2, y aquí está el binario 01 0000, que es 16 en decimal



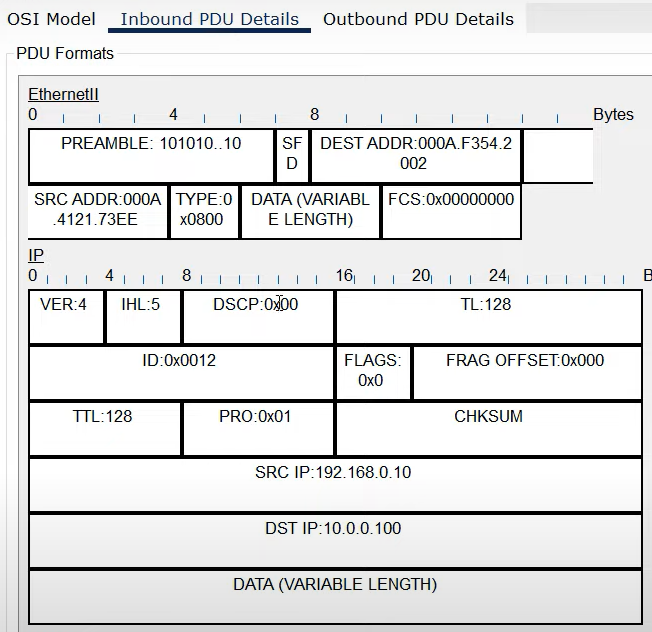
## Parte 5: Envíe tráfico HTTP de la PC1 al servidor SRV1 y verifique las marcas.

**Paso 1: Abra el modo de simulación y realice un acceso por web al servidor SRV1.**

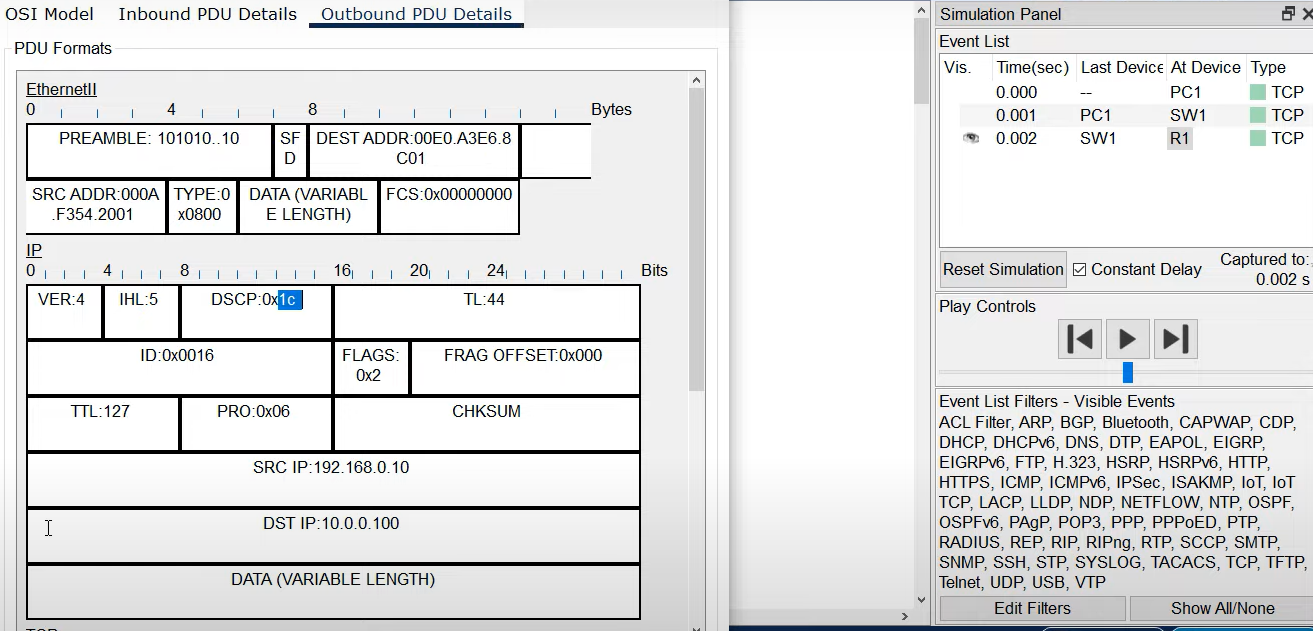
**http://10.0.0.100**

Llevemos el paquete a R1 y luego verifiquemos.

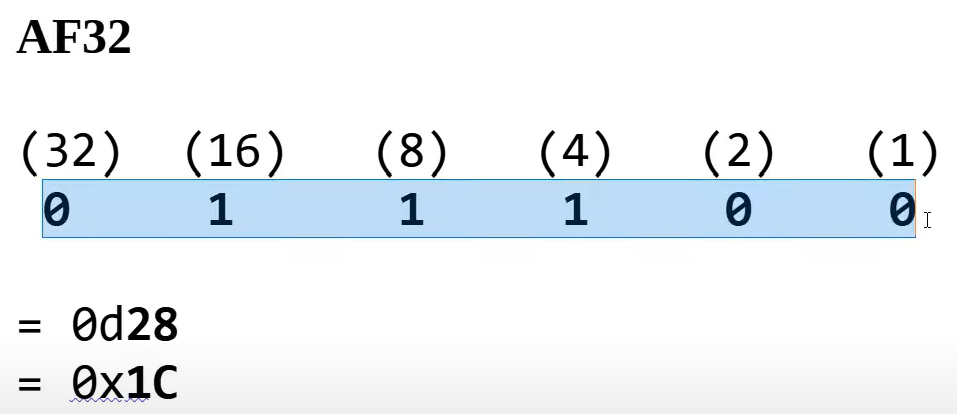
* En la pestaña de **inbound PDU details**, la marca de DSCP es 0.



* En la pestaña de **outbound PDU details**, la marca de DSCP es 1 C.



* Entonces, lo marcamos como AF32, revisemos las matemáticas



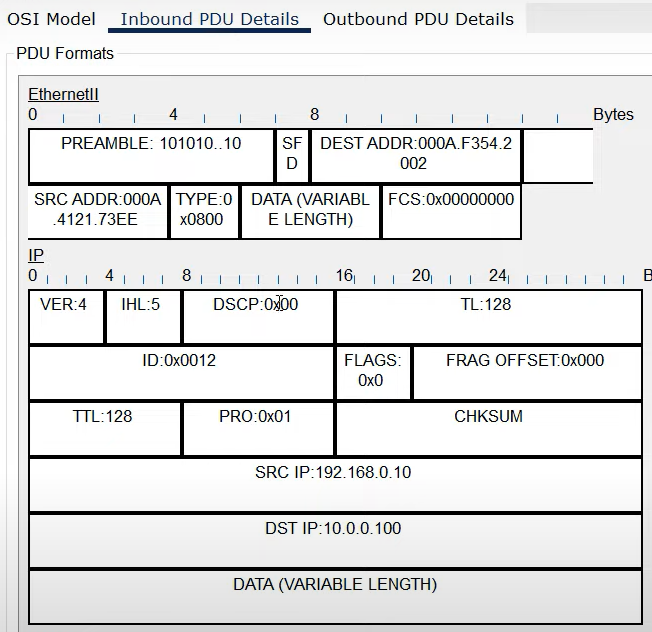
## Parte 5: Envíe tráfico HTTPS de la PC1 al servidor SRV1 y verifique las marcas.

**Paso 1: Abra el modo de simulación y realice un acceso por web al servidor SRV1.**

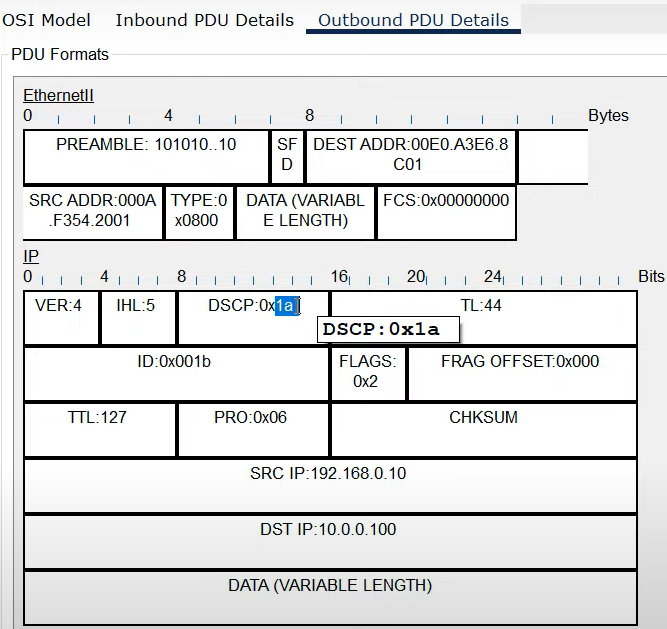
**https://10.0.0.100**

Llevemos el paquete a R1 y luego verifiquemos.

* En la pestaña de **inbound PDU details**, la marca de DSCP es 0.

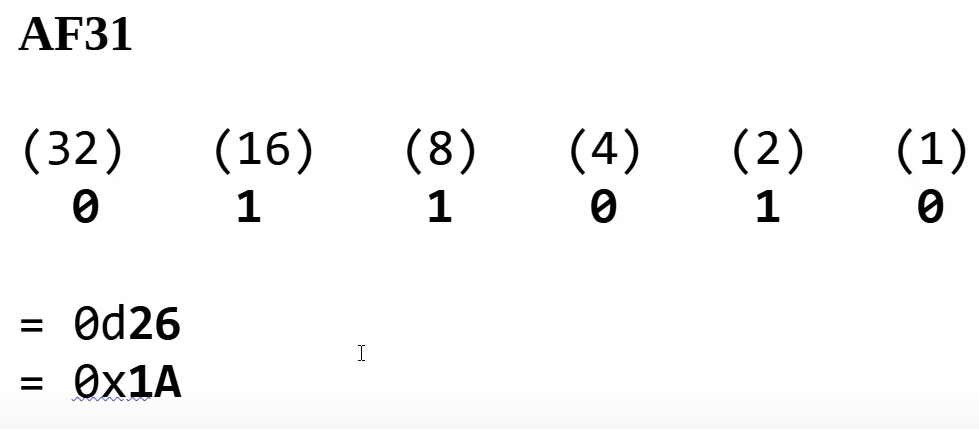


* En la pestaña de **outbound PDU details**, la marca de DSCP es 1A.



* R1 lo ha marcado como DSCP 1A, que debería ser AF31. Revisemos las

Matemáticas:



Los mapas de clases identifican el tráfico (class maps)

Los mapas de políticas (policy maps) especifican qué acciones tomar en ese tráfico

Las políticas de servicio aplican los mapas de políticas a las interfaces.

* Entonces, lo marcamos como CS2, y aquí está el binario 01 0000, que es 16 en decimal

