**Cuestionario 2**

**Direcciones IP**

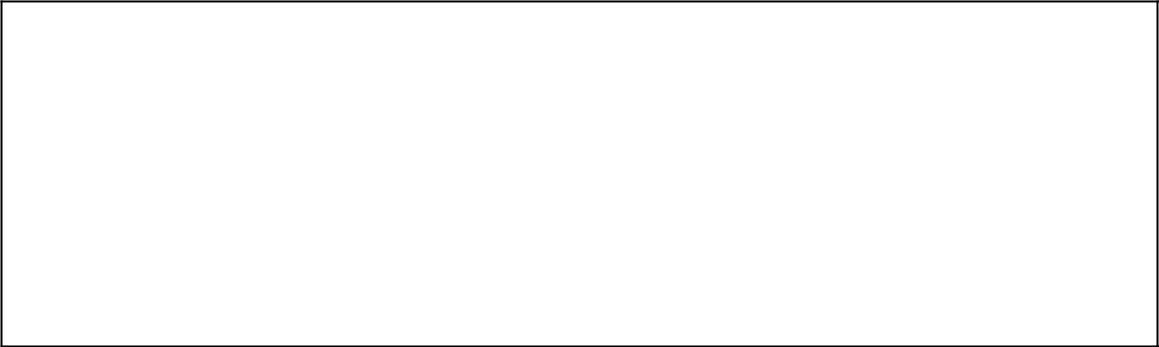
Día de Cursada:

Sede:

Docente a cargo:

Fecha de Inicio:

Grupo



· Imprima este PDF

· Responda a mano en los lugares indicados

· Escanee el documento o tomo un foto de buena calidad

· Pase a PDF

· Suba su TP a [trabajopractico@gmail.com](mailto:trabajopractico@gmail.com) y Google Drive.



El TP es grupal y se entrega como máximo a los 14 días de iniciado

1. Indique todos los tamaños posibles **del encabezado** (Header) de un datagrama. ( Valor

mínimo, valor máximo y valores intermedios)

El tamaño del encabezado de un datagrama puede variar dependiendo del protocolo de red que estés utilizando. A continuación, te proporciono algunos ejemplos de tamaños de encabezado de datagrama para diferentes protocolos:

1. \*\*IPv4 (Internet Protocol versión 4)\*\*:

- Valor mínimo: 20 bytes (sin opciones ni datos).

- Valor máximo: 60 bytes (con muchas opciones y sin datos).

- Valores intermedios: Los tamaños típicos de encabezado IPv4 oscilan entre 20 y 40 bytes, dependiendo de las opciones y configuraciones específicas.

2. \*\*IPv6 (Internet Protocol versión 6)\*\*:

- Valor mínimo: 40 bytes (sin extensiones ni datos).

- Valor máximo: 60 bytes (con extensiones y sin datos).

- Valores intermedios: Los tamaños de encabezado IPv6 suelen variar según las extensiones utilizadas y pueden ser mayores que 60 bytes en casos excepcionales.

3. \*\*TCP (Transmission Control Protocol)\*\*:

- Valor mínimo: 20 bytes (sin opciones ni datos).

- Valor máximo: 60 bytes (con muchas opciones y sin datos).

- Valores intermedios: El tamaño típico del encabezado TCP varía entre 20 y 60 bytes, dependiendo de las opciones y configuraciones específicas.

4. \*\*UDP (User Datagram Protocol)\*\*:

- Valor mínimo: 8 bytes (sin datos).

- Valor máximo: 8 bytes (sin datos).

- Valores intermedios: El encabezado UDP tiene un tamaño fijo de 8 bytes, independientemente de los datos que se transmitan.

Estos son ejemplos de tamaños de encabezado para algunos protocolos comunes en redes. Ten en cuenta que estos tamaños pueden variar en función de las opciones específicas de configuración y las extensiones utilizadas en cada protocolo.

2. Suponga un archivo de **100 Kbyte** que se transmite directamente sobre UDP (considere

UDP con un formato de payload y header que se encapsula sobre IP, el Header UDP es de **8 Bytes**). Se sabe que el origen y el destino están conectados mediante **2 Routers en** **serie**.

· **MTU** host origen a router1**: 1500 Bytes**

· **MTU** router 1 a router 2 : **500 Bytes**

**·** **MTU** router 2 a host destino : **1500 Bytes** Averigüe:

1. Cantidad de Bytes que llegan al destino si no hay errores

2. Cantidad de paquetes que circulan en cada enlace

3 Tamaño de paquete IP en cada enfade

4. Contenido del campo offset del último paquete de cada enlace.

Para determinar la cantidad de bytes que llegan al destino, la cantidad de paquetes en cada enlace, el tamaño de los paquetes IP en cada enlace y el contenido del campo de desplazamiento (offset) del último paquete en cada enlace, debemos dividir el archivo de 100 Kbytes en paquetes y tener en cuenta las restricciones de MTU en cada enlace. Vamos a calcular esto paso a paso:

Supongamos que el archivo de 100 Kbytes se divide en paquetes de tamaño máximo permitido en cada enlace. Dado que el MTU (Maximum Transmission Unit) es el tamaño máximo de paquete que se puede transmitir en una red sin fragmentar, tenemos lo siguiente:

MTU host origen a router 1: 1500 bytes

MTU router 1 a router 2: 500 bytes

MTU router 2 a host destino: 1500 bytes

1) Cantidad de Bytes que llegan al destino si no hay errores:

- Enlace 1 (host origen a router 1): El tamaño del paquete IP será de 1500 bytes (MTU del enlace). Como el encabezado UDP es de 8 bytes, el tamaño de los datos (payload) en cada paquete será de 1500 - 8 = 1492 bytes.

- Enlace 2 (router 1 a router 2): El tamaño del paquete IP será de 500 bytes (MTU del enlace). Como el encabezado UDP es de 8 bytes, el tamaño de los datos en cada paquete será de 500 - 8 = 492 bytes.

- Enlace 3 (router 2 a host destino): El tamaño del paquete IP será de 1500 bytes (MTU del enlace). Como el encabezado UDP es de 8 bytes, el tamaño de los datos en cada paquete será de 1500 - 8 = 1492 bytes.

Para calcular la cantidad de paquetes, dividimos el tamaño del archivo (100 Kbytes) entre el tamaño de los datos en cada paquete:

- Enlace 1: 100 Kbytes / 1492 bytes por paquete ≈ 67 paquetes.

- Enlace 2: 100 Kbytes / 492 bytes por paquete ≈ 203 paquetes.

- Enlace 3: 100 Kbytes / 1492 bytes por paquete ≈ 67 paquetes.

3) Tamaño de paquete IP en cada enlace:

- Enlace 1: 1500 bytes (MTU del enlace).

- Enlace 2: 500 bytes (MTU del enlace).

- Enlace 3: 1500 bytes (MTU del enlace).

4) Contenido del campo offset del último paquete de cada enlace:

- Enlace 1: Dado que hay 67 paquetes en total y asumiendo que la numeración comienza desde 0, el último paquete tendrá un campo de desplazamiento (offset) de 66 \* 1492 bytes = 98,312 bytes.

- Enlace 2: Dado que hay 203 paquetes en total, el último paquete tendrá un campo de desplazamiento (offset) de 202 \* 492 bytes = 99,384 bytes.

- Enlace 3: Dado que hay 67 paquetes en total, el último paquete tendrá un campo de desplazamiento (offset) de 66 \* 1492 bytes = 98,312 bytes.

En resumen:

1) Cantidad de Bytes que llegan al destino si no hay errores: 100 Kbytes.

2) Cantidad de paquetes que circulan en cada enlace:

- Enlace 1: 67 paquetes.

- Enlace 2: 203 paquetes.

- Enlace 3: 67 paquetes.

3) Tamaño de paquete IP en cada enlace:

- Enlace 1: 1500 bytes.

- Enlace 2: 500 bytes.

- Enlace 3: 1500 bytes.

4) Contenido del campo offset del último paquete de cada enlace:

- Enlace 1: 98,312 bytes.

- Enlace 2: 99,384 bytes.

- Enlace 3: 98,312 bytes.

3. **IP** es un protocolo no confiable, no orientado a la conexión.¿ Es posible hacer una redconfiable y orientada a la conexión que corra sobre IP ? ¿Cómo?

Sí, es posible construir una red confiable y orientada a la conexión que utilice el Protocolo de Internet (IP) como base. Aunque IP en sí mismo es un protocolo no confiable y no orientado a la conexión, es común utilizar otros protocolos y técnicas en la capa superior para lograr la confiabilidad y la orientación a la conexión. Aquí hay algunas formas de lograrlo:

1. \*\*TCP (Transmission Control Protocol)\*\*: TCP es un protocolo de capa de transporte que se ejecuta sobre IP y proporciona confiabilidad y orientación a la conexión. TCP establece una conexión entre dos puntos finales, realiza un seguimiento del orden de los datos y garantiza que los datos se entreguen de manera confiable y en orden. También maneja la detección y retransmisión de paquetes perdidos o dañados.

2. \*\*UDP con Control de Aplicación\*\*: Aunque UDP es no confiable por naturaleza, las aplicaciones pueden agregar su propio control de confiabilidad y orientación a la conexión en la capa de aplicación. Esto significa que las aplicaciones pueden implementar sus propios mecanismos de retransmisión, control de flujo y detección de errores para garantizar la confiabilidad de los datos transmitidos sobre UDP.

3. \*\*Túneles VPN (Virtual Private Network)\*\*: Las redes privadas virtuales pueden implementarse sobre IP para crear conexiones confiables y seguras entre redes geográficamente separadas. Los protocolos VPN, como IPSec, establecen túneles seguros que encapsulan los datos IP y proporcionan confidencialidad, autenticación e integridad de los datos transmitidos.

4. \*\*Protocolos de Aplicación Personalizados\*\*: En algunos casos, las aplicaciones pueden implementar sus propios protocolos de capa de aplicación para garantizar la confiabilidad y la orientación a la conexión. Estos protocolos personalizados pueden incluir características como confirmación de entrega, retransmisión y control de flujo según las necesidades específicas de la aplicación.

En resumen, aunque IP en sí mismo es un protocolo no confiable y no orientado a la conexión, es posible lograr la confiabilidad y la orientación a la conexión en una red que utiliza IP como base mediante el uso de protocolos de capa superior como TCP o la implementación de mecanismos personalizados en la capa de aplicación. Esto permite a las aplicaciones y servicios de red funcionar de manera confiable sobre la infraestructura IP existente.

4. El **protocolo ARP** averigua la **dirección MAC** del destino, enviando broadcast, para luego usando esa dirección enviar los mensajes en unicast. ¿Porque no se envía todo por broadcast y evitamos el ARP?

La razón principal por la que no se envían todos los mensajes de una red por broadcast en lugar de utilizar el Protocolo ARP (Address Resolution Protocol) para obtener la dirección MAC del destino es la eficiencia y la reducción del tráfico de red innecesario. Aquí hay varias razones por las cuales el ARP es preferible a enviar todo por broadcast:

1. \*\*Eficiencia en el uso del ancho de banda\*\*: El broadcast envía un paquete a todas las estaciones en la red, lo que significa que todas las estaciones deben procesar y descartar el paquete si no es para ellas. Esto desperdicia ancho de banda y capacidad de procesamiento en las estaciones que no son el destinatario real.

2. \*\*Escalabilidad\*\*: A medida que una red crece en tamaño, la cantidad de tráfico de broadcast también aumenta exponencialmente. Esto puede sobrecargar la red y causar congestión. El ARP permite una resolución de direcciones específica, lo que minimiza el tráfico innecesario.

3. \*\*Seguridad\*\*: El broadcast de todos los paquetes en una red expone la información a todas las estaciones, lo que puede ser un riesgo de seguridad si se transmiten datos sensibles. El ARP se utiliza para determinar la dirección MAC del destinatario específico, lo que limita quién recibe los datos.

4. \*\*Reducción de la latencia\*\*: Enviar paquetes directamente al destinatario conocido a través de unicast es más rápido que enviarlos a través de broadcast, donde todos deben procesarlos y descartarlos. Esto reduce la latencia y mejora el rendimiento de la red.

5. \*\*Eficiencia en la administración de recursos\*\*: El ARP se utiliza para mantener una tabla de asignaciones de direcciones IP a direcciones MAC en los dispositivos de red. Esto permite una administración más eficiente de los recursos y una resolución de direcciones más rápida en el futuro, ya que no es necesario realizar broadcast repetidos para la misma dirección IP.

En resumen, el ARP se utiliza para obtener la dirección MAC del destino de manera eficiente y específica en una red, lo que minimiza el tráfico innecesario y mejora el rendimiento, la seguridad y la escalabilidad de la red en comparación con el uso de broadcast generalizado.

5. Los paquetes se segmentan cuando **el MTU** de la red no es suficiente, al llegar al destino se rearman. Analice la posibilidad, ventajas y desventajas de rearmarlo en ruteadores intermedios.

Segmentar y rearmar paquetes en los enrutadores intermedios a lo largo de la ruta de la red es una técnica que se utiliza en algunos casos, pero tiene sus ventajas y desventajas. Aquí se analizan las posibilidades, ventajas y desventajas de realizar esta operación en los enrutadores intermedios:

\*\*Posibilidades\*\*:

1. \*\*Segmentación en Ruteadores Intermedios\*\*:

- Enrutadores intermedios pueden dividir paquetes grandes en fragmentos más pequeños cuando el MTU de la red siguiente es menor que el tamaño del paquete.

- Cada fragmento se envía por separado a través de la red y, en el destino, se deben rearmar para obtener el paquete original.

\*\*Ventajas\*\*:

1. \*\*Mejora la compatibilidad\*\*: Permite que los dispositivos finales que no son compatibles con la segmentación de paquetes (por ejemplo, algunos dispositivos IoT o sistemas embebidos) puedan comunicarse a través de redes con diferentes MTUs.

2. \*\*Menos fragmentación en redes posteriores\*\*: Al realizar la segmentación en el enrutador intermedio, se evita que los dispositivos en redes posteriores tengan que realizar la fragmentación, lo que puede simplificar su procesamiento y reducir la probabilidad de pérdida de fragmentos.

\*\*Desventajas\*\*:

1. \*\*Aumento de la sobrecarga de procesamiento en los enrutadores\*\*: La segmentación y el reensamblaje de paquetes requieren recursos de procesamiento en los enrutadores intermedios. En redes con un tráfico intenso y muchos paquetes grandes, esto puede aumentar la carga de trabajo de los enrutadores y disminuir su rendimiento.

2. \*\*Posibilidad de fragmentación excesiva\*\*: Si se realiza segmentación en múltiples enrutadores intermedios a lo largo de la ruta, un paquete grande podría fragmentarse en numerosos fragmentos más pequeños. Esto podría aumentar la complejidad del reensamblaje en el destino y aumentar la probabilidad de pérdida de fragmentos.

3. \*\*Riesgo de ataques de fragmentación\*\*: Los ataques de fragmentación pueden aprovecharse de la segmentación y el reensamblaje de paquetes para crear tráfico malicioso. Los enrutadores deben implementar mecanismos de seguridad para evitar este tipo de ataques.

4. \*\*Aumento de la latencia\*\*: La segmentación y el reensamblaje pueden agregar una pequeña cantidad de latencia a la transmisión de datos, ya que se requiere tiempo adicional para realizar estas operaciones en cada enrutador intermedio y en el destino.

En resumen, la segmentación y el reensamblaje de paquetes en enrutadores intermedios pueden ser útiles en ciertos escenarios para mejorar la compatibilidad y reducir la fragmentación en redes posteriores. Sin embargo, también introducen sobrecarga de procesamiento, aumentan la complejidad y pueden aumentar la latencia, por lo que deben implementarse con cuidado y considerarse en función de las necesidades y las limitaciones de la red específica.

6. Un paquete de **5000 Bytes** con **14 Bytes** de cabecera Ethernet, **20 Byte**s de cabecera IP y **20 Byte** de cabecera UDP se fragmenta para pasar por una red Ethernet de **MTU 1514** **Bytes.** Indique para cada fragmento:

a. Long Total

b. Indicador de mas fragmentos

c. Desplazamiento ( Offset)

Para fragmentar un paquete de 5000 bytes para que pueda pasar por una red Ethernet con un MTU de 1514 bytes, debemos calcular la cantidad y el tamaño de los fragmentos. Aquí están los detalles para cada fragmento:

Tamaño total del paquete original: 5000 bytes

Tamaño de la cabecera Ethernet: 14 bytes

Tamaño de la cabecera IP: 20 bytes

Tamaño de la cabecera UDP: 20 bytes

Tamaño de datos en el paquete original: 5000 - 14 - 20 - 20 = 4946 bytes

MTU de la red Ethernet: 1514 bytes

Ahora calculamos la cantidad y el tamaño de los fragmentos:

1. \*\*Fragmento 1\*\*:

- Tamaño total: 1514 bytes

- Indicador de más fragmentos: 1 (ya que habrá más fragmentos)

- Desplazamiento (offset): 0 bytes

2. \*\*Fragmento 2\*\*:

- Tamaño total: 1514 bytes

- Indicador de más fragmentos: 1 (ya que habrá más fragmentos)

- Desplazamiento (offset): 1514 bytes (porque es el primer byte del segundo fragmento)

3. \*\*Fragmento 3\*\*:

- Tamaño total: 1514 bytes

- Indicador de más fragmentos: 1 (ya que habrá más fragmentos)

- Desplazamiento (offset): 3028 bytes (porque es el primer byte del tercer fragmento)

4. \*\*Fragmento 4\*\*:

- Tamaño total: 1514 bytes

- Indicador de más fragmentos: 1 (ya que habrá más fragmentos)

- Desplazamiento (offset): 4542 bytes (porque es el primer byte del cuarto fragmento)

5. \*\*Fragmento 5\*\*:

- Tamaño total: 1340 bytes (los bytes restantes)

- Indicador de más fragmentos: 0 (este es el último fragmento)

- Desplazamiento (offset): 6056 bytes (porque es el primer byte del quinto fragmento)

En resumen, el paquete original se fragmenta en cinco fragmentos para que pueda pasar por una red Ethernet con un MTU de 1514 bytes. Cada fragmento tiene un tamaño total de 1514 bytes, excepto el último fragmento que tiene 1340 bytes. El indicador de más fragmentos se establece en 1 para todos los fragmentos, excepto para el último que se establece en 0. El desplazamiento (offset) varía para cada fragmento para indicar la posición de los datos en el paquete original.

7. Suponga que un **paquete IP** atraviesa un Ruteador para conectar dos redes IDENTICAS en todos sus parámetros. ¿Cambia en algo el paquete IP?

Si un paquete IP atraviesa un enrutador para conectar dos redes idénticas en todos sus parámetros, el contenido del paquete IP, incluyendo los datos y las cabeceras IP, generalmente no debería cambiar. La función principal de un enrutador es reenviar paquetes entre redes y tomar decisiones de enrutamiento en función de la dirección de destino IP, pero no suele modificar el contenido del paquete IP a menos que se apliquen políticas específicas de red o transformaciones específicas.

Aquí hay algunas consideraciones clave:

1. \*\*Cabecera IP\*\*: La cabecera IP, que contiene información esencial como la dirección de origen y la dirección de destino IP, generalmente no se modifica en un enrutador a menos que sea necesario para el enrutamiento, por ejemplo, cuando se cambia la dirección de origen para que sea la dirección IP del enrutador.

2. \*\*Datos del paquete\*\*: Los datos transportados por el paquete IP, conocidos como el campo de carga útil o payload, no deberían cambiar a menos que se apliquen transformaciones específicas, como la traducción de direcciones de red (NAT) o la inspección profunda de paquetes (DPI).

3. \*\*Otras cabeceras\*\*: Si el paquete IP transporta protocolos de nivel superior, como TCP o UDP, las cabeceras de estos protocolos también deberían permanecer intactas, a menos que se realicen modificaciones específicas en el enrutador.

4. \*\*Fragmentación y reensamblaje\*\*: Si el paquete IP es demasiado grande para la MTU (Maximum Transmission Unit) de la red de destino, el enrutador podría fragmentarlo en fragmentos más pequeños. Sin embargo, esto no suele implicar cambios en el contenido del paquete, solo en su estructura para que se pueda transmitir a través de la red de destino. El reensamblaje de los fragmentos se realiza en el destino final.

En resumen, en un escenario en el que un paquete IP atraviesa un enrutador para conectar dos redes idénticas en todos sus parámetros, el paquete IP en sí mismo generalmente no cambia su contenido, a menos que se requieran modificaciones específicas para el enrutamiento, la seguridad o la gestión de la red. El enrutador principalmente se encarga de tomar decisiones de enrutamiento y reenviar el paquete de manera eficiente.

8. Dos máquinas **A** y **B** están interconectadas mediante el ruteador **R**. Indique para cada salto: MAC origen; MAC destino; IP Origen; IP destino

Para indicar las direcciones MAC y las direcciones IP en cada salto entre las máquinas A, B y el enrutador R, necesitamos saber más detalles sobre la configuración de las direcciones IP y las tablas de enrutamiento en cada dispositivo. Sin esta información, no es posible proporcionar direcciones MAC e IP específicas para cada salto.

Sin embargo, puedo proporcionarte un ejemplo genérico para ilustrar cómo podría ser el proceso:

Supongamos una topología de red simple con las siguientes direcciones:

- Máquina A:

- Dirección IP: 192.168.1.2

- Dirección MAC: AA:BB:CC:DD:EE:FF

- Máquina B:

- Dirección IP: 192.168.1.3

- Dirección MAC: 11:22:33:44:55:66

- Ruteador R:

- Interfaz hacia A:

- Dirección IP: 192.168.1.1

- Dirección MAC: 00:11:22:33:44:55

- Interfaz hacia B:

- Dirección IP: 192.168.1.4

- Dirección MAC: 66:77:88:99:AA:BB

El proceso de comunicación podría ser el siguiente:

\*\*Salto 1 (Máquina A a Ruteador R):\*\*

- MAC origen: AA:BB:CC:DD:EE:FF (de A)

- MAC destino: 00:11:22:33:44:55 (del ruteador R)

- IP origen: 192.168.1.2 (de A)

- IP destino: 192.168.1.1 (del ruteador R)

\*\*Salto 2 (Ruteador R a Máquina B):\*\*

- MAC origen: 66:77:88:99:AA:BB (del ruteador R)

- MAC destino: 11:22:33:44:55:66 (de B)

- IP origen: 192.168.1.4 (del ruteador R)

- IP destino: 192.168.1.3 (de B)

Esto es solo un ejemplo genérico, y las direcciones MAC y las direcciones IP en una red real dependerán de la configuración de esa red en particular. Cada dispositivo en la red tendrá su propia dirección MAC y dirección IP, y las tablas de enrutamiento determinarán cómo se encaminan los paquetes de un dispositivo a otro a través de la red.

9. Suponga un enlace vía cable de **256 Kbps** entre dos esta estaciones distanciadas **3000 Km**. El tiempo de ping con paquete de **64 Bytes** es de **36 ms**. Estime si el enlace está sobrecargado.

Para determinar si el enlace está sobrecargado, primero debemos calcular la capacidad teórica del enlace en términos de ancho de banda y luego compararla con el rendimiento observado en función del tiempo de ping.

La capacidad teórica de un enlace de 256 Kbps significa que puede transmitir hasta 256 kilobits por segundo.

Sin embargo, la distancia entre las dos estaciones también afecta al tiempo de propagación de la señal (latencia). Para calcular la latencia teórica, podemos utilizar la fórmula de retardo de ida y vuelta (Round-Trip Time, RTT):

RTT = 2 \* distancia / velocidad de la luz

Donde:

- RTT es el tiempo de ping redondeado (36 ms en este caso).

- La distancia entre las estaciones es de 3000 Km (o 3,000,000 metros).

- La velocidad de la luz es aproximadamente 299,792,458 metros por segundo.

Sustituyendo estos valores en la ecuación:

36 ms = 2 \* (3,000,000 m) / velocidad de la luz

Ahora, resolvemos para la velocidad de la luz:

Velocidad de la luz = 2 \* (3,000,000 m) / (36 ms) ≈ 166,666,666.67 metros por segundo

Dado que la velocidad de la luz es constante, podemos concluir que el tiempo de propagación de la señal en esta conexión es de aproximadamente 36 ms.

Ahora, vamos a calcular cuántos bits pueden viajar desde una estación a la otra durante este tiempo:

Capacidad teórica del enlace = 256 Kbps = 256,000 bps

Duración de la señal (en segundos) durante el RTT de 36 ms = 0.036 segundos

Bits transmitidos durante el RTT = Capacidad teórica del enlace \* Duración del RTT

Bits transmitidos durante el RTT = 256,000 bps \* 0.036 segundos ≈ 9,216 bits

Dado que durante el RTT se pueden transmitir aproximadamente 9,216 bits, y estás usando paquetes de 64 bytes (512 bits), podemos calcular cuántos paquetes se podrían transmitir durante el RTT:

Número de paquetes transmitidos durante el RTT = Bits transmitidos durante el RTT / Tamaño del paquete

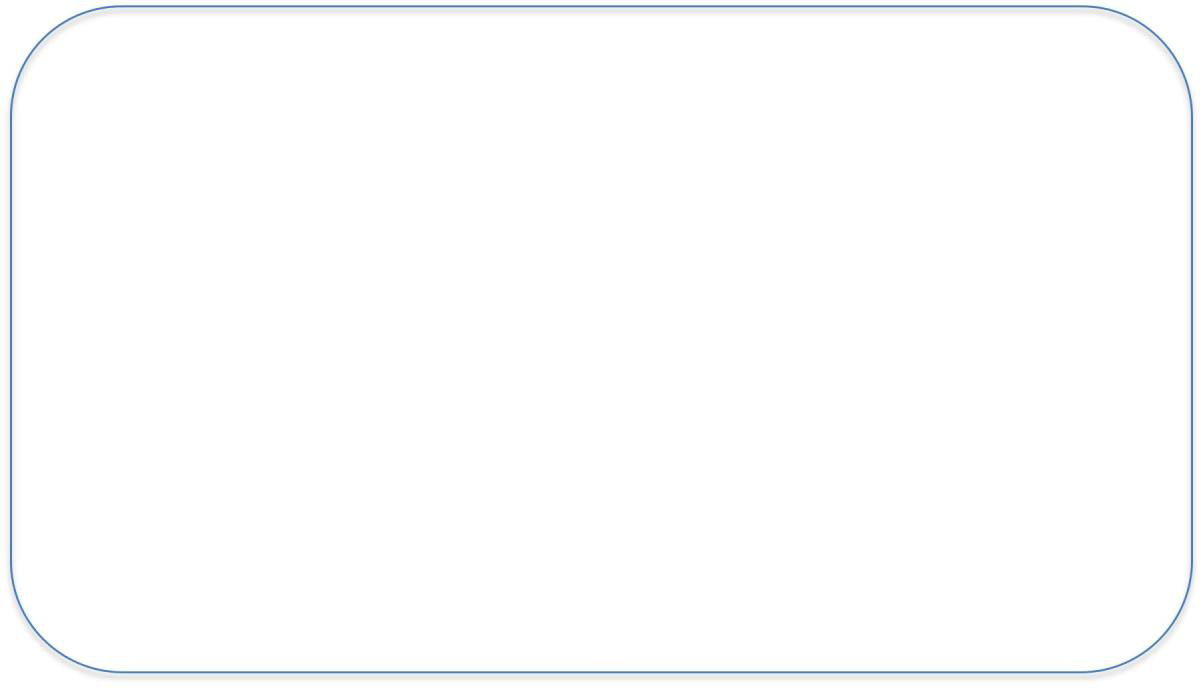
Número de paquetes transmitidos durante el RTT = 9,216 bits / 512 bits ≈ 18 paquetes

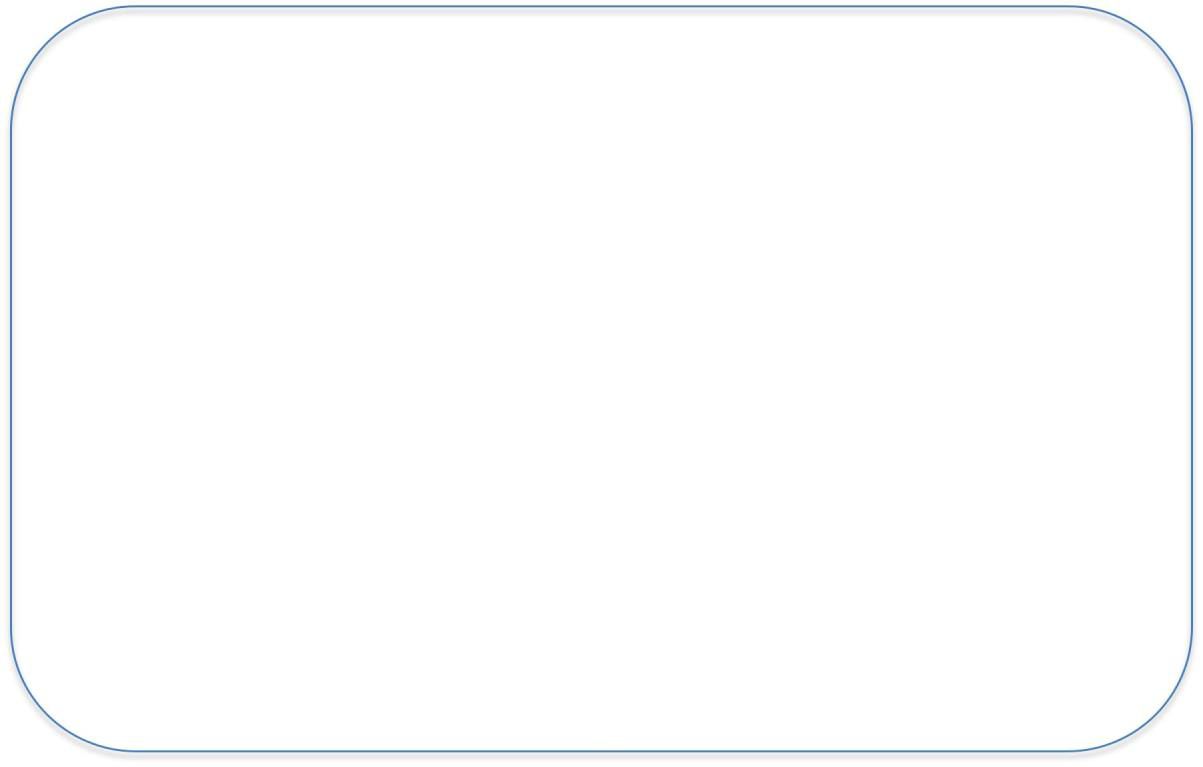
En este escenario, se pueden transmitir aproximadamente 18 paquetes de 64 bytes durante el RTT. Si estás viendo un tiempo de ping de 36 ms, que es el tiempo de ida y vuelta, y solo se pueden transmitir 18 paquetes durante ese tiempo, podría ser un indicio de que el enlace está cerca de su capacidad máxima. Sin embargo, para una evaluación más precisa de la congestión, es importante considerar otros factores, como la carga de la red en ese momento y la posible congestión en el enlace.

10. Calcule la **máxima tasa de bits** a la cual un nodo puede generar paquetes de **100 Bytes** para evitar que fragmentos de distintos paquetes se puedan re ensamblar juntos (equivocadamente).

Suponga que el máximo tiempo de vida de un paquete en la red sea de **60 seg** antes de ser descartados

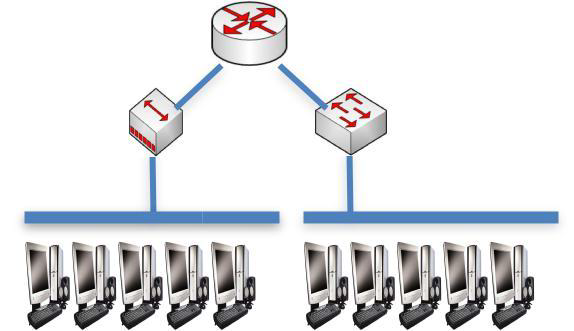
*AYUDA: Tome en cuenta que los paquetes tienen identificadores que se incrementan cada vez que se envía un paquete, cuando se llega al valor máximo se vuelve a reiniciar la cuenta.*



11. Diferencie **dominio de Broadcast** de **Dominio de Colisión.** **Ejemplifique.** 

12: ¿Cuantos **dominios de Broadcast** y cuantos de **colisión** se muestran en el gráfico?

Explique.



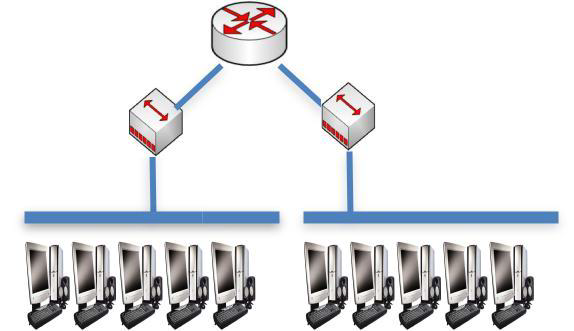
13. Indique con cuales de los siguientes dispositivos genera distintos dominios de **Broadcast** y cuales distintos **dominios de colisión**. Indique brevemente que función cumple cada uno.

a. Repetidores.

b. Hub

c. Switch

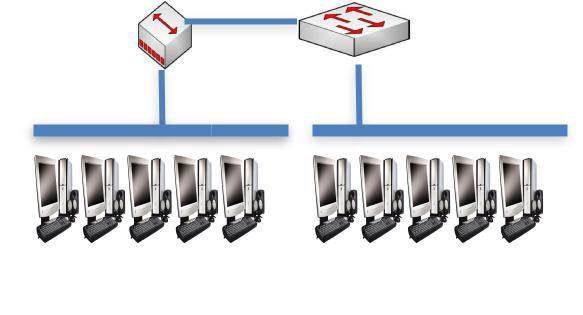
d Routers

14 . Para la siguiente red indique cuantos **dominios de Broadcast** y cuantos **dominios de** **colisión** hay? 

15. Los administradores suelen segmentar sus redes. ¿Qué ventaja obtienen?



16. ¿Qué host pueden **transmitir simultáneamente** sin provocar colisiones?



1 2 3 4 5 6 7 8 9 10

17. ¿Cuantas **direcciones de Host** se pueden obtener de una red **clase C**?

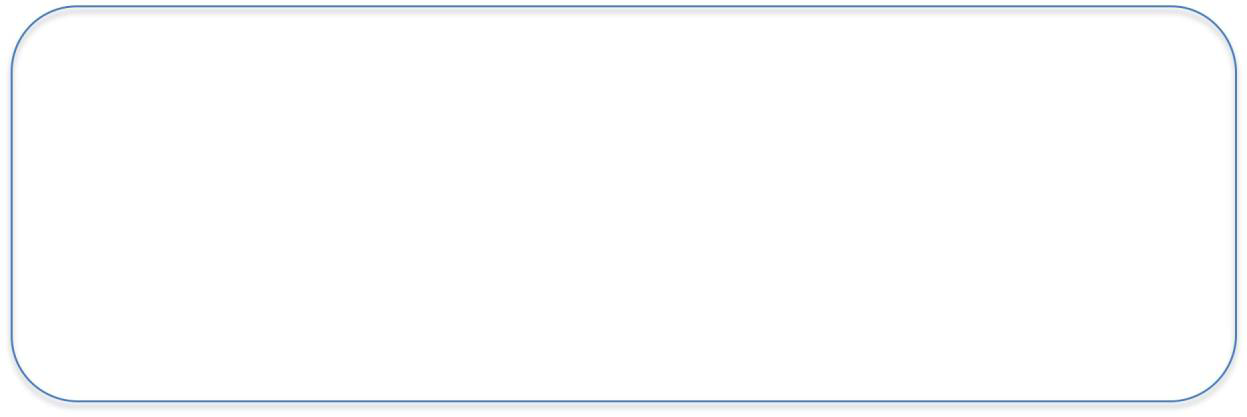


18 Demuestre que **un fragmento IP** DEBE ser múltiplo de **8 Bytes**. 

19 Suponga un datagrama de **tamaño máximo** que se fragmenta para pasar por una red IP de **mínimo MTU** (576 bytes). Indique cual será el contenido del campo de offset del último  
fragmento.



20. Diferencie claramente el significado de dirección **Unicast**, **Multicast** y **Anycast** en IPv6



21. Escriba la dirección IPv6 completa de 2001:DB8::202:B3FF:FE1E:8329



22. Una forma de trabajar con IPv4 en IPv6 es usar la dirección IPv4 como los cuatro últimos octetos de IPv6. Sabiendo esto a que dirección IPv6 corresponde. [**192.168.0.2**](http://192.168.0.2)