PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2333 — Sistemas Operativos y Redes — 1/2019
Examen (Redes)

Jueves 27-Junio-2019

Duración: 3 horas
SIN CALCULADORA

1. **[11p]** Responda verdadero o falso. Las falsas solo reciben puntaje si están justificadas.

   Verdaderas: 0 si es incorrecta; 1 si es correcta.

   Falsas: 0 si es incorrecta, no hay justificación, o la justificación es incoreca; 1 si la justificación es correcta.

   1.1) Un *switch* puede conectarse con otros *switch*, pero se debe tener cuidado de no formar conexiones circulares (ciclos) entre ellos.

   Falso. Las conexiones circulares entre *switch* sí son posibles. El protocolo STP evita los problemas de la existencia de ciclos.

   No es suficiente (0p) decir que los *switches* sí pueden conectarse circularmente. Para que sea considerado correcto debe decir que al menos que 'existe un protocolo que soluciona el problema de los ciclos'.

   1.2) En el modelo basado en capas, el protocolo de aplicación del *host* de destino recibe debe descartar los *headers* agregados en las capas inferiores.

   Falso. En el *host* de destino, cada una de las capas inferiores elimina los *headers* que le conciernen. El protocolo de aplicación en el destino recibe el mensaje de la misma manera que se lo envió el *host* de origen. No recibe un mensaje con *header* de transporte.

   1.3) El protocolo DNS es necesario para encontrar el camino a un destinatario en la red

   Falso. El problema de encontrar un destinatario en la red es de la capa de red. DNS es un protocolo de capa de aplicación que permite asociar un nombre lógico (texto) a un dirección de capa de red. Una red sin DNS pueden funcionar igualmente, aunque talvez haya alguna incomodidad más a nivel de aplicación.

   1.4) A nivel de transporte, el puerto es una manera única de identificar a un proceso dentro de un *host*.

   Verdadero. La interpretación es que el puerto permite identificar de manera única a un proceso. Si dicen que es falso, tendrían que explicar por qué lo están interpretando así.

   1.5) Para probar un programa distribuido entre dos procesos en un único computador, es necesario que este computador utilices dos direcciones IPv4

   Falso. Los procesos pueden abrir sockets en puertos distintos dentro del mismo computador, y utilizar la dirección de *loopback* 127.0.0.1 (o bien 0.0.0.0).

   La respuesta debería mencionar al menos el uso de puertos distintos.

   1.6) En una red IPv4, los paquetes pueden pasar por múltiples subredes.

   Verdadero.

   1.7) Dentro de una red IPv4 **privada** no es necesario que todos los miembros tengan direcciones distintas.

   Falso. Dentro de cada subred (privada o pública) las direcciones deben ser distintas.

   1.8) Al usar NAT, el paquete IPv4 se modifica tanto al salir de la red interna, como al entrar a ella.

   Verdadero.

   1.9) En ausencia de un servidor DHCP en la red, las entradas de una tabla ARP deben ser configuradas manualmente para que un *host* pueda comunicarse con otros miembros de la red.

   Falso. La tabla ARP se configura de manera automática.

1.10) En redes inalámbricas, si un miembro conectado a un *access point* no detecta transmisiones de otros miembros, entonces puede transmitir sin problemas.

Falso. Esto no es cierto por el problemas de los terminales ocultos. Se puede mencionar que las redes inalámbricas usan CSMA/CA, o que el *access point* usa paquetes especiales RTS y CTS para indicar quién puede transmitir.

1.11) Protocolos de capa de enlace como *ethernet* utilizan direcciones de *hardware* (direcciones MAC) para encontrar a un *host*.

Verdadero. De hecho la dirección MAC es parte del *header ethernet*

2. **[16p]** Responda de manera breve y lo más precisa posible las siguiente preguntas.

2.1) **[4p]** El protocolo TCP utiliza el concepto de números de secuencia dentro del *header* de un segmento. ¿Por qué son necesarios los números de secuencia para el funcionamiento del protocolo y cómo los define TCP?

R. 2p. El segmento y *header* por si solo no basta para identificar de manera única a un dato enviado por la red. El número de secuencia permite distinguir si se trata del mismo segmento o no.

2p. TCP define los números de secuencia como la cantidad de byte que se han recibido correctamente. También se podría decir que indica el número del siguiente byte a recibir.

2.2) **[4p]** En una subred IPv4, cuando un *host* desea enviar un mensaje, hay una decisión importante que debe tomar: "¿está el destinatario en la misma subred que yo?".

a) ¿Cómo determina el *host* si el destinatario se encuentra en la misma subred que él?

b) ¿Por qué esa pregunta es importante?

R. 2p. Comparando el resultado del AND entre la máscara y la IP tanto del origen como del destino https://www.overleaf.com/project/5d14f0116a81013c73b78fd8 2p. Si es la misma subred el destinatario debería encontrarse en la red local (y puede preguntarle al switch o encontrarlo via ARP). En caso contrario, debe hacer llegar el mensaje al *router*, para que él lo reenvíe a la subred correcta. En ese caso, la dirección MAC que va dentro del *header ethernet* (o WiFi) debe ser la dirección MAC del *gateway*.

2.3) **[4p]** A nivel de capa de red, hay dos maneras de transmitir un mensaje: la versión *connection-oriented* que utiliza **circuitos virtuales**, y la versión *connection-less*, que utiliza **datagramas**. La comunicación tradicional telefónica usa circuitos virtuales, mientras que Internet usa el modelo de datagramas. En cuanto a la información que necesita para tomar una decisión de ruteo, ¿qué diferencia hay entre un *router* para circuitos virtuales, y un *router* para datagramas?

R.

2p. El *router* de circuitos virtuales necesita asociar un número de circuito virtual de entrada con un número de circuito virtual de salida. No necesita que el paquete a enviar contenga dirección de origen ni dirección de destino.

2p. El *router* de datagramas almacena direcciones de destino, y las salidas asociadas a cada una de ella. Necesita analizar un datagrama para extraer la dirección de destino y tomar la decisión.

2.4) **[4p]** En la comunicación entre dos *hosts* a nivel de capa de aplicación, al momento de utilizar *sockets* TCP, la API indica que para enviar un mensaje se debe utilizar el comando `send`, el cual **NO incluye** la dirección de destino del mensaje. Por otro lado, cuando se usan *sockets* UDP, se recurre al comando `sendto`, que sí incluye una estructura con el destinatario del mensaje. ¿A qué se debe esta diferencia de diseño entre ambos comandos? La misma situación es válida entre los respectivos `recv` y `recvfrom`.

R.

2p. *Sockets* TCP requieren estar en un estado `CONNECTED`, luego de haber ejecutado `accept` y `connect`. Esto permite establecer los datos de origen y destino como parte del canal de comunicación, por lo tanto no es necesario utilizar especificarlos en cada envío.

2p. *Sockets* UDP no establecen conexión previa. Por esto se necesario que cada operación de envío y recepción contenga los datos de *host* que se encuentra en el otro extremo.

3. **[15p]** Considere el segmento IPv4 10.10.31.128/X, y complete la siguiente tabla.

	$X = 20$	$X = 24$	$X = 28$
Máscara de subred			
Dirección *broadcast*			
Número máximo de *hosts*			
Menor IP posible en la subred			
Mayor IP posible en la subred			

	$X = 20$	$X = 24$	$X = 28$
Máscara de subred	255.255.240.0	255.255.255.0	255.255.255.240
Dirección *broadcast*	10.10.31.255	10.10.31.255	10.10.31.143
Número máximo de *hosts*	4094	254	14
Menor IP posible en la subred	10.10.16.1	10.10.31.1	10.10.31.129
Mayor IP posible en la subred	10.10.31.254	10.10.31.254	10.10.31.142

4. **[18p]** Suponga que posee un laptop nuevo, con tarjeta de red inalámbrica, y desea conectarse a una red WiFi para acceder por primera vez al contenido de la página `http://iic2333.ing.puc.cl/index.html`. El servidor se encuentra dentro de la red pública cableada del DCC, la cual es distinta a la red usada por la WiFi.

   En base a los contenidos del curso, describa la mayor cantidad de pasos y protocolos involucrados en cada etapa. Su respuesta debe incluir al menos un protocolo por cada capa entre aplicación, transporte, red y enlace, pero algunas capas involucran la participación de más protocolos.

   Tome como referencia la siguiente tabla:

Capa	Elemento	Protocolo	Acción
Número o nombre	*Host* origen, *host* destino, *router*, *switch*, ...	*Nombre*	*descripción*

   Procure describir los pasos de manera ordenada. Sin embargo si descubre que necesita pasos intermedios, escríbalos al final y agregue números al costado para indicar el orden de manera clara.

   Pregunta bastante abierta. Al menos se espera que se mencione DHCP, DNS, HTTP, Ethernet, IP, ARP, TCP

Capa	Elemento	Protocolo	Acción
Enlace	*Host* origen, access point	WiFi	Conectarse a red WiFi
Red	*Host* origen, servidor	DHCP	Obtener configuración IP
Aplicación	*Host* origen	DNS	Obtener IP de `iic2333.ing.puc.cl`
Transporte	*Host* origen	UDP	Enviar mensaje a servidor DNS
Red	*Host* origen	IP	Encontrar ubicación servidor DNS mediante IP
Enlace	*Host* origen, *switch*	ARP	Obtener MAC de servidor DNS (no la encuentra)
Enlace	*Host* origen, *gateway*	WiFi	Obtener MAC de gateway y enviar mensaje
Red	*gateway*, *routers*	IP	Enviar paquetes a subred de destino
Transporte	*Host* origen, destino DNS	UDP	Enviar mensaje a servidor DNS
Aplicación	*Host* origen, destino DNS	DNS	Respuesta con IP de `iic2333.ing.puc.cl`
Aplicación	*Host* origen	HTTP	Envío de GET para `index.html`
Transporte	*Host* origen, destino HTTP	TCP	Envío de mensaje `SYN` (*handshake*)
Red	*Host* origen, *gateway*	IP	Enviar mensaje via *gateway*
Enlace	*Host* origen, *switch*	WiFi	Enviar mensaje a *gateway*
Red	*gateway*, *routers*	IP	Enviar paquetes a subred de destino
Enlace	*gateway* destino, *switch*	Ethernet	Enviar paquetes a destino
Red	*host* destino	IP	Pasar paquetes a capa superior
Transporte	*host* destino	TCP	Recibir paquete `SYN`, completar *handshake*
Transporte	*host* origen	TCP	Enviar mensaje HTTP
Red/Enlace	origen,*switch*,*routers*, dest	WiFi/IP/Ethernet	Enviar mensaje a destino
Red	*gateway*, *routers*	IP	Hacer llegar mensaje a destino
Transporte	*host* destino	TCP	Recibir paquetes, verificar integridad, retransmisión, etc
Aplicación	*host* destino	HTTP	Recibir solicitud y responder

Criterios:

- Deben notarse 4 etapas: (1) conexión a la WiFi, (2) obtención de IP con DHCP, (3) obtención de IP de destino con DNS, (4) envío de solicitud HTTP
- 9p para las 4 etapas, distribuidos como:
  - 1p. Para etapa (1). Existencia y hacer mención a protocolo WiFi.
  - 2p. Para etapa (2). Existencia y hacer mención a uso de DHCP. No puede haber otros mensajes enviadas a la red antes de esta etapa.
  - 3p. Para etapa (3). Existencia. Protocolo de transporte puede ser TCP o UDP.
  - 3p. Para etapa (4). Existencia. Basta que haya una solicitud. No es necesario mostrar la respuesta y los sucesivos mensajes pidiendo imágenes ó CSS.
- 2p. Debe haber un *handshake* TCP mencionado (para HTTP). No es necesario que estén todos los mensajes asociados al proceso de *handshake*
- 2p. Debe notarse que, para DNS y HTTP, los mensajes parten en la capa de aplicación, bajan hasta red/enlace, y vuelven a subir hasta aplicación en el destino.
- 2p. Transporte y aplicación deben estar asociados solamente a *host* origen y/o *host* destino
- 2p. Debe haber un proceso de obtención de MAC usando ARP, y que algunos mensajes pasan a través del *gateway*
- 1p. Debe mencionar el protocolo *ethernet* alguna vez, por ejemplo, para llegar al servidor HTTP.

5. **[15p]** Esta pregunta es **OPCIONAL**. Si la responde, su puntaje reemplazará a la peor pregunta del *midterm* incondicionalmente (aunque sea más baja).

   Considere un sistema de archivos indexado con bloques de tamaño B byte, y cada puntero a un bloque es de 4 byte. El nodo índice utiliza P punteros de direccionamiento directo a bloques de datos, S punteros de indirección simple, y D punteros de indirección doble.

   5.1) ¿Cuántos datos puede almacenar un archivo que utiliza únicamente los punteros de direccionamiento directo?
   5p. $P \times B$

   5.2) ¿Cuántos datos puede almacenar un archivo que utiliza únicamente los punteros de indirección simple?
   5p. $S \times B/4 \times B$

   5.3) ¿Cuántos datos puede almacenar un archivo que utiliza únicamente los punteros de indirección doble?
   5p. $D \times B/4 \times B/4 \times B$

i	2^i B
6	64 B
7	128 B
8	256 B
9	512 B
10	1024 B

i	2^i B
10	1 KB
20	1 MB
30	1 GB
40	1 TB