

Solución Examen – 24 de julio de 2023**Instrucciones**

- Indique su nombre completo y número de cédula en cada hoja.
- Numere todas las hojas e indique la cantidad total de hojas que entrega en la primera.
- Escriba las hojas de un solo lado y utilice una caligrafía claramente legible.
- Comience cada pregunta teórica y cada ejercicio en una hoja nueva.
- Solo se responderán dudas de letra. No se responderán dudas de ningún tipo los últimos 30 minutos del examen.
- El examen es individual y sin material. Apague su teléfono celular mientras esté en el salón del examen.
- Es obligatorio responder correctamente al menos 15 puntos en las preguntas teóricas y 20 de los problemas prácticos. Los puntos ganados en el curso se suman a los puntos de teórico.
- El puntaje mínimo de aprobación es de 60 puntos.
- Para todos los ejercicios, si es necesario, puede suponer que dispone de los tipos de datos básicos (p.ej. lista, cola, archivo, string, etc.) y sus funciones asociadas (ej: tail(lista), crear(archivo), concatenar(string, string)).
- Justifique todas sus respuestas.
- Duración: 3 horas. Culinadas las 3 horas el alumno no podrá modificar las hojas a entregar de ninguna forma.

Preguntas Teóricas**Pregunta 1 (8 puntos)**

- Se pretende configurar un nuevo dominio de DNS, con nombre `redes202307.net`. Si el servidor que será el autoritativo para dicho dominio tiene la dirección IP `192.5.6.31` y el nombre `ns.redes202307.net`, indique qué registros se deben configurar y en qué servidor/es para activar la delegación de su administración.
- Si se desea disponer también de la administración del dominio `redes202307.net.uy` también en el servidor con IP `192.5.6.31`, ¿es necesario realizar alguna configuración adicional?
- Se configurará un único servidor web asociado a los nombres `www.redes202307.net` y `www.redes202307.net.uy`, el que tendrá la dirección IP `192.5.6.32`. ¿Qué registro/s hay que agregar y dónde para que el mismo sea alcanzable desde cualquier sitio de Internet?

Solución Pregunta 1

a) Tanto en el servidor con IP `192.5.6.31` (por ser el autoritativo para dicha zona) como en los servidores autoritativos de la zona "net" (para realizar la delegación de la administración), se debe configurar lo siguiente:

```
redes202307.net      IN      NS      ns.redes202307.net
```

formato libro: (redes202307.net, ns.redes202307.net, NS)

```
ns.redes202307.net  IN      A       192.5.6.31
```

formato libro: (ns.redes202307.net, 192.5.6.31, A)

b) Sí, se debe configurar en la zona `redes202307.net.uy` del servidor `192.5.6.31` así como también en los servidores autoritativos de la zona "net.uy" (para realizar la delegación de la administración)

```
redes202307.net.uy  IN      NS      ns.redes202307.net.uy
```

```
ns.redes202307.net.uy IN      A       192.5.6.31
```

Vale formato libro

c) Se debe agregar en la zona `redes202307.net`

```
www.redes202307.net  IN      A       192.5.6.32
```

y en la zona `redes202307.net.uy`

```
www.redes202307.net.uy IN      A       192.5.6.32
```

Vale formato libro

Pregunta 2 (8 puntos)

Al utilizar UDP es posible enviar mensajes de Broadcast simplemente indicando la dirección IP correspondiente. Sin embargo, en TCP no existe el concepto de Broadcast. Explique la razón, indicando que conceptos de la implementación de TCP son incompatibles con este tipo de servicio.

Solución Pregunta 2

TCP es orientado a conexión, y depende de variables de estado que describen la configuración propia (p.ej. números de secuencia) y del otro extremo de la conexión (p.ej. último ACK recibido). Estas variables se usan para implementar los servicios de TCP tales como entrega confiable y control de flujo. Estas variables son transmitidas en los segmentos TCP en campos dedicados. Resumiendo:

- El protocolo TCP no tiene previsto la posibilidad de que haya más de dos participantes en una conexión.
- TCP depende de un establecimiento de conexión para inicializar el estado de los terminales antes de empezar a transmitir datos
- La implementación de los servicios TCP dependen de conocer el estado de los nodos participantes, lo que entra en contradicción con la idea de enviarles a todos incluso sin saber a quiénes.

Pregunta 3 (8 puntos)

- ¿Cómo se codifican y representan las direcciones de capa 2 (MAC) y las de la capa 3 (IPv4 e IPv6)?
- ¿Qué instituciones asignan los valores de las direcciones en ambos casos?
- ¿Porqué se dice que las direcciones de capa 3 son jerárquicas?

Solución Pregunta 3

a) Las direcciones MAC son de tamaño 48 bits, y se representan en una secuencia de 6 pares de números hexadecimales separados por dos puntos (:). Un ejemplo es 1A:2F:BB:76:09:AD

Las IP son en caso de IPv4 32 bits y se representan con cuatro cifras entre 0 y 255 (representación de un byte), separados por punto (.). Un ejemplo es 200.36.12.25

Por último IPv6 son direcciones con tamaño de 128 bits. Se representan en hexadecimal, agrupado de a 4 dígitos y separados por ::. Un ejemplo es 2001:0db8:85a3:0000:0000:8a2e:0370:7334, también notada como 2001:db8:85a3::8a2e:370:7334

b) La asignación de direcciones MAC es administrada por IEEE. Por lo tanto, los fabricantes compran porciones del espacio de direcciones MAC (asegurando su unicidad). La dirección se compone de dos elementos, OUI (Organizationally Unique Identifier), que son los tres primeros octetos (company_id) y los restantes 3 octetos administrados por la compañía.

Las direcciones IP para dispositivos conectados a Internet tiene la exigencia que ésta dirección IP debe ser única para toda la Red.

Las direcciones son entregadas por bloques a los IPS que distribuyen entre sus clientes. Los bloques son administrados por ICANN: Internet Corporation for Assigned Names and Numbers, que tiene entre otras exigencias reparto de direcciones, manejo del DNS y asignación de nombres de dominios. Además se encarga de resolver disputas.

Asimismo hay organismos que se encargan de gestionar dichas asignaciones regionalmente. Estas organizaciones son Arin, Ripe, Apnic, Lacnic y Afrinic.

c) Las direcciones de capa 3 se denominan jerárquicas, por tener en la dirección una componente que representa la subred a la que pertenece y otra que identifica el host dentro de la subred. Los primeros bits indican la subred y los últimos indican el elemento en la subred.

El direccionamiento jerárquico permite ser más eficiente en la publicación de la información de ruteo, dado que permite agregar las redes y publicar menos información.

Pregunta 4 (8 puntos)

a) Describa el atributo AS_PATH de BGP, y explique su utilidad en:

- i) enrutamiento inter-dominio (eBGP)
- ii) enrutamiento intra-dominio (iBGP)

b) Considere un ISP que tiene dos proveedores de nivel superior con los que mantiene sesiones BGP. El ISP en cuestión gestiona el prefijo 22.22.0.0/23; ¿cómo debe anunciarlo por BGP a sus proveedores para intentar lograr balanceo de carga y tolerancia a fallos ante la caída de un enlace?

Solución Pregunta 4

a) El atributo AS_PATH contiene los números de AS (ASNs) a través de los cuales se ha transmitido el anuncio de un determinado prefijo. Cuando un router recibe un anuncio de un prefijo desde un vecino BGP externo, agrega su ASN al AS_PATH antes de difundirlo a otros vecinos BGP.

i) En eBGP, si un router recibe un anuncio de prefijo que contiene su propio ASN en el atributo AS_PATH, sabe que existe un ciclo y debe descartar el anuncio.

ii) En iBGP, el AS_PATH no se puede usar para detectar ciclos dentro del AS, ya que el ASN no cambia en este caso.

En ambos casos, el largo del atributo AS_PATH se utiliza como métrica en el proceso de decisión de BGP

b) Se debe anunciar:

En el primer enlace

22.22.0.0/24 → se prefiere que el tráfico hacia la mitad “baja” del prefijo completo venga por este enlace

22.22.0.0/23 → respaldo

En el segundo enlace

22.22.0.1/24 → se prefiere que el tráfico hacia la mitad “alta” del prefijo completo venga por este enlace

22.22.0.0/23 → respaldo

Pregunta 5 (8 puntos)

Explique en qué consiste el problema de conteo a infinito en protocolos de enrutamiento de vector distancia. Comente posibles técnicas para mitigar este problema.

Solución Pregunta 5

El problema del conteo infinito se genera por el propio mecanismo utilizado por los protocolos vector-distancia, ya que si bien los nodos reciben los vectores-distancias de sus vecinos, que ellos luego utilizan para re-calculan el propio, no tienen conocimiento de si alguno de esos costos corresponden a caminos que pasan por ellos y por ende es posible que se generen bucles de enrutamiento. Cuando uno tiene visión global de la red es un problema sencillo de detectar y evitar, pero en los algoritmos vector-distancia solo se maneja información parcial.

El horizonte dividido (split-horizon) es uno de los métodos utilizados por los protocolos de enrutamiento de vector de distancia para evitar bucles de enrutamiento. El principio es simple: un enrutador no anunciará una ruta de regreso a la interfaz desde la cual se aprendió.

Otro método para solucionar éste problema es la reversa envenenada (route poisoning) donde se realiza una modificación del algoritmo de vector distancia, evitando que un nodo informe a su vecino sobre la distancia que conoce hasta el nodo X cuando la trayectoria hacia X pasa a través de ese nodo vecino. Lo que realiza es informar que dicha distancia es infinito.

Problema 1 (30 puntos)

La empresa “RecienArranco” necesita diseñar la red interna para los puestos de trabajo con miras en el crecimiento. Dado que no cuenta con presupuesto para adquirir un rango de direcciones públicas, contrata al proveedor de servicio de confianza un plan de servicio básico. Este proveedor le instala un router con funcionalidad NAT y le configura la IP 191.100.4.2/30 en su interfaz hacia Internet. Este router solo cuenta con dos interfaces de red en total.

La empresa tendrá 3 áreas de trabajo: Ventas, Administración y Desarrollo. Hoy en día cuentan con 6 puestos de trabajo en Ventas, 4 en Administración y 10 en Desarrollo. Se espera que en dos años Ventas crezca un 100%, Administración otro 100% y Desarrollo un 300%.

Los usuarios de las áreas de Desarrollo y Administración deben contar con conectividad a Internet, pero en ningún caso deben poder ser contactados desde Internet.

Ningún equipo en ningún área debe poder ver tráfico interno del resto de las áreas. Además, por temas de seguridad los usuarios de Ventas **sólo** deben poder comunicarse con usuarios de Administración.

Se pide:

a) Proponga un diseño de la red, detallando los dispositivos que utilice y las tecnologías que implementen. Debe justificar cada uno de ellos en base a los requerimientos de la empresa. Tenga en cuenta que para esta tarea puede adquirir equipos nuevos, pero recuerde que el presupuesto es acotado, por lo cual, su solución debe ser eficiente.

b) Numere todas la/s subred/es de su diseño, de manera que se ajuste a las necesidades del problema.

c) Escriba las tablas de forwarding/reenvío de todos los dispositivos que lo requieran para cumplir con los requerimientos de comunicación planteados. Para los casos en que la tabla sea exactamente la misma, alcanza con que escriba solo una y mencione para quienes sería igual.

d) Cada vez que inicia sesión un host en el área de Administración se realiza automáticamente el envío de un formulario mediante un POST HTTP, con el usuario y la hora del inicio de sesión, al servidor HTTP 1.0 con IP 121.8.0.10.

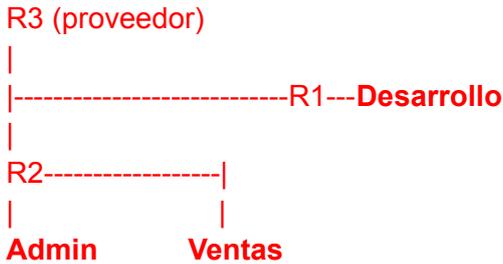
Muestre todos los paquetes entrantes y salientes observados en la interfaz hacia Internet, luego de que un operario cualquiera en el área de Administración inicia sesión en su puesto de trabajo. Para esta tarea puede asumir que solo hay tráfico relacionado a dicho procedimiento y que los enlaces dentro del ISP utilizan la tecnología Ethernet.

Debe utilizar el siguiente formato, donde “Cabezal Transporte” debe estar con los campos más relevantes del protocolo que utilice.

MAC_o | MAC_d | IP_o | IP_d | Cabezal Transporte | payload | justificación

Solución Problema 1

a) Para cumplir con las restricciones se propone el siguiente diseño:



Para esto, se deben adquirir dos routers: uno para la subred del área de Desarrollo (R1) y otro para Administración y Ventas (R2).

El router R1 es necesario para cumplir el requerimiento de no comunicación entre Ventas y Desarrollo, si se utilizara solo un router las rutas DC me habilitarían dicha comunicación.

En el router R2 se pueden configurar dos subredes utilizando dos interfaces diferentes para las subredes de Administración y Ventas (que se pueden comunicar entre si). Luego, para impedir que los hosts de Ventas no accedan a Internet no se puede dejar en manos de este router omitiendo la ruta por defecto, pues de esta manera los hosts de Administración tampoco accederían a Internet. Para no utilizar un dispositivo más, se debe configurar un servidor DHCP en R2 que solo le distribuya la red DC de Administración a la subred de Ventas. Otra opción es que un administrador configure los hosts de Ventas sin la ruta por defecto.

Finalmente, para completar el acceso a Internet se debe configurar adecuadamente la tabla de forwarding de R3.

b)

Subred entre routers: 3 Ips + red + broadcast → 10.1.0.0/29

Subred Desarrollo: 10 + 30 hosts + ifR1 + red + broadcast = 43 → 10.2.0.0/26

Subred Ventas: 6 + 6 hosts + ifR2 + red + broadcast = 15 → 10.3.0.0/28

Subred Administración: 4 + 4 hosts + ifR2 + red + broadcast = 11 → 10.4.0.0/28

c)

R1

Prefijo	Netx Hop	Interfaz
10.1.0.0/29	DC	eth0
10.2.0.0/26	DC	eth1
10.4.0.0/28	10.1.0.2	eth0
0/0	10.1.0.3	eth0

R2

Prefijo	Netx Hop	Interfaz
10.1.0.0/29	DC	eth0
10.2.0.0/26	10.1.0.1	eth0
10.3.0.0/28	DC	eth1
10.4.0.0/28	DC	eth2
0/0	10.1.0.3	eth1

R3

Prefijo	Netx Hop	Interfaz
10.1.0.0/29	DC	eth1
10.2.0.0/26	10.1.0.1	eth1
10.4.0.0/28	10.1.0.2	eth1
191.100.4.2/30	DC	eth0
0/0	191.100.4.1	eth0

Host Venta (todos guales)

Prefijo	Netx Hop	Interfaz
10.3.0.0/28	DC	eth0
10.4.0.0/28	10.3.0.1	eth0

Host Administración (el resto son iguales)

Prefijo	Netx Hop	Interfaz
10.4.0.0/28	DC	eth0
0/0	10.4.0.1	eth0

Host Desarrollo (el resto son iguales)

Prefijo	Netx Hop	Interfaz
10.2.0.0/26	DC	eth0
0/0	10.2.0.1	eth0

d) Como hay otros hosts de otras áreas funcionando y no se especifica el estado de las tablas, los mensajes ARP pueden obviarse. Además, al tener la IP destino no necesitamos usar DNS. Además, se asume que no hay pérdidas de paquetes y que el el POST HTTP entra en un segmento de datos.

MAC_o	MAC_d	IP_o	IP_d	Cabecal Transporte	payload	justificación
MAC_R3eth0	s/d	191.100.4.2	121.8.0.10	Puerto_o: 1234 Puerto_d: 80 num_sec: 0 num_ack: 0 Flag SYN	-	Solicitud de conexión TCP
s/d	MAC_R3eth0	121.8.0.10	191.100.4.2	Puerto_o: 80 Puerto_d: 1234 num_sec: 0 num_ack: 0 Flag SYN, ACK	-	Respuesta a solicitud de conexión TCP
MAC_R3eth0	s/d	191.100.4.2	121.8.0.10	Puerto_o: 1234 Puerto_d: 80 num_sec: 0	POST HTTP usuario: user hora: 8:00	Ultimo ACK del 3-way handshake + datos

Redes de Computadoras

				num_ack: 0 Flag ACK	
s/d	MAC_R3eth0	121.8.0.10	191.100.4.2	Puerto_o: 80 - Puerto_d: 1234 num_sec: 0 num_ack: 120	ACK de los datos recibidos
s/d	MAC_R3eth0	121.8.0.10	191.100.4.2	Puerto_o: 80 - Puerto_d: 1234 num_sec: 0 num_ack: 120 Flag FIN	Inicio de cierre de conexión
MAC_R3eth0	s/d	191.100.4.2	121.8.0.10	Puerto_o: - 1234 Puerto_d: 80 num_sec: 120 num_ack: 0 Flag ACK	ACK para el FIN
MAC_R3eth0	s/d	191.100.4.2	121.8.0.10	Puerto_o: - 1234 Puerto_d: 80 num_sec: 120 num_ack: 0 Flag FIN	FIN del cliente
s/d	MAC_R3eth0	121.8.0.10	191.100.4.2	Puerto_o: 80 - Puerto_d: 1234 num_sec: 0 num_ack: 120 Flag ACK	ACK para el FIN

Problema 2 (30 puntos)

Sea un sistema de streaming que funciona de la siguiente manera:

El programa cliente inicia y se conecta por TCP al servidor cuyo FQDN es *stream.fing.edu.uy* y solicitará recibir el stream enviando el comando:

```
CONECTAR <PUERTO_UDP_CLIENTE>\n
```

indicando el puerto UDP en el que recibirá los fotogramas. A medida que recibe los fotogramas, los irá reproduciendo en pantalla mediante la invocación a una función (ya implementada) *reproducirFotograma()*.

El servidor estará escuchando por conexiones TCP al puerto 4444, a la espera de clientes que quieran recibir el stream (que envían el comando CONECTAR), y publicará la señal de video enviando los fotogramas por UDP desde el puerto 5555 a todos los clientes que lo solicitaron. El servidor obtiene los fotogramas mediante la invocación a una función (ya implementada) *obtenerFotograma()*, que retorna el fotograma listo para ser usado como mensaje en un send de UDP.

Nota: asuma que no debe manejar el cierre del programa cliente, a lo sumo si del lado del servidor detecta que la conexión se cerró, puede cerrar y eliminar ese cliente de la lista.

Se pide:

Implemente en un lenguaje de alto nivel, usando la API de sockets del curso el servidor que atiende el servicio y cliente que lo solicita.

Solución Problema 2**SERVIDOR DE STREAM****def serverStream()**

```
# Para mituoexcluir acceso al diccionario
```

```
mutex = semaphore()
```

```
# Diccionario para agregar las duplas IP:PUERTO de los clientes activos
```

```
Cientes = new Diccionario()
```

```
#creo socket para envio de video por UDP
```

```
sktUdp = socket.udp()
```

```
sktUdp.bind(*, 5555)
```

```
#creo socket para atender clientes por TCP
```

```
server = socket.tcp()
```

```
server.bind(*, 4444)
```

```
server = server.listen()
```

```
# Inicio hilo de envío de stream
```

```
new.thread(enviarStream, sktUdp)
```

```
# Espero nuevos clientes
while true:
    client, err = server.accept()
    new.thread(atenderCliente, client)
end

server.close()
sktUdp.close()
```

end

def enviarSream(socket stream)

```
while true do

    # Espero fotograma
    fotograma=obtenerFotograma()

    # Bloqueo la lista de clientes
    mutex.acquire()

    # para cada cliente activo, se le envia fotograma
    for k, v in Clientes.items():
        # envio mensaje UDP con el fotograma
        aux = v.split(':')
        # en la posicion 0 tengo la IP y en la 1 el puerto
        stream.sendto(fotograma, aux[0], aux[1])

    # Libero la lista de clientes
    mutex.release()
```

end

end

```

def atenderCliente(socket cmd)
  msg = ""
  buff = ""
  IP_PORT = 0 # IP:Puerto UDP donde recibe el video

  // espero solicitud con numero de puerto
  repeat
    r, err = client.receive()
    if err=="closed" then return end
    buff = buff + r
    // busco primer \n en datos recibidos
    pos = string.pos(buff, "\n")
  until (pos > 0)

  dato = buff.split(' ')

  # En la posicion 0 esta el comando y en la 1 el puerto
  if ( dato[0] == 'CONECTAR' ) then
    ip, port = client.getpeer() # obtengo la ip del cliente
    IP_PORT = ip+':'+dato[1] # obtengo el puerto del parametro

    # Bloqueo la lista de clientes
    mutex.acquire()

    # asumo que si existe, no hace nada
    Clientes.add(IP_PORT,IP_PORT)

    # Libero la lista de clientes
    mutex.release()

  end

  # ya no tengo mas que hacer, que esperar un desconexion
  # o error de conexion
  repeat
    sleep(30)
    r, err = client.receive()
  until err=="closed" or err=nil

  # Elimino el cliente de la lista y cierro el socket
  mutex.acquire()
  Clientes.remove(IP_PORT)
  mutex.release()
  client.close()

```

end

end

CLIENTE DE STREAM**def cliente()**

```
#creo socket TCP para enviar comandos al servidor
```

```
comandos = socket.tcp()
```

```
#creo socket para recibir video por UDP
```

```
sktUdp = socket.udp()
```

```
ip, port = sktUdp.gethost()
```

```
sktUdp = sktUdp.listen()
```

```
# Me conecto al servidor
```

```
repeat
```

```
    clientCmd, err = comandos.connect(stream.fing.edu.uy, 4444)
```

```
until err!="closed" and err!=nil
```

```
# envio CONECTAR
```

```
msg= 'CONECTAR $port\n'
```

```
repeat
```

```
    msg, err = client.send(msg)
```

```
until (msg==" or err=='closed')
```

```
if err then
```

```
    comandos.close()
```

```
    sktUdp.close()
```

```
    return
```

```
end
```

```
# Ahora ya me registre en el server para escuchar el stream
```

```
while true do
```

```
    fotograma, ip, port = skt.receive()
```

```
    reproducirFotograma(fotograma)
```

```
end
```

```
comandos.close()
```

```
sktUdp.close()
```

```
end
```