

Redes de Computadoras  
**Solución – 27 de julio 2017**  
(ref: solredes20170727.odt)

### 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.
- Sólo 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. Culminadas las 3 horas el alumno no podrá modificar las hojas a entregar de ninguna forma.

### Preguntas Teóricas

#### Pregunta 1 (10 puntos)

- a) Explique qué es un medio compartido y cite al menos dos ejemplos.
- b) Describa las clases de protocolos de acceso al medio compartido y cite un ejemplo para cada una de dichas clases de protocolos.

#### Solución

a) Es un medio de difusión, donde muchos emisores pueden transmitir simultáneamente, generando colisiones con determinada probabilidad. Un ejemplo de medio compartido es el aire, o medio inalámbrico, otro ejemplo es el cable coaxial donde se conectan múltiples nodos.

b) Las clase de protocolos son i) "por partición de canal", ii) de "acceso aleatorio" y iii) por "toma de turnos". En el caso i) se divide el medio "en porciones", por ejemplo en fracciones de frecuencias del medio electromagnético, o en fracciones de tiempo o "time-slots". En el caso ii) se puede citar el protocolo de acceso en ethernet, denominado "Carrier Sense Multiple Access/ Collision Detection" (CSMA/CD), y en el caso iii) un caso típico es "Token Ring".

#### Pregunta 2 (6 puntos)

En el contexto de multicast, explique en qué consiste el mecanismo de "Reverse Path Forwarding", y para qué sirve.

#### Solución

"Reverse Path Forwarding" (RPF) es un mecanismo de inundación (*flooding*) controlado, que permite determinar por qué enlaces se va a re-enviar un mensaje de difusión. Cuando un node recibe un mensaje desde una fuente, lo reenviará por todos sus enlaces excepto por donde llegó, solo si dicho enlace está en el camino más corto desde el nodo a la fuente. RPF sirve para implementar inundación controlada, como se dijo anteriormente.

#### Pregunta 3 (7 puntos)

Explique mediante un ejemplo sencillo por qué es posible generar oscilaciones en el enrutamiento cuando se utilizan costos dinámicos para los enlaces, proporcionales al tráfico cursado por ellos.

#### Solución

Usaremos el ejemplo del libro que refiere a protocolos "Link State", pero el problema puede aparecer en otros tipos de protocolos. Dado un "rombo" con un enrutamiento inicial en sentido antihorario (figura a), si se inicia un tráfico "e" desde y hacia x, se encontrará un camino de menor costo en sentido horario, que como resultado recargará el enlace z-w (figura b), determinando un nuevo cambio de sentido. Esta situación oscilará permanentemente.

## Redes de Computadoras

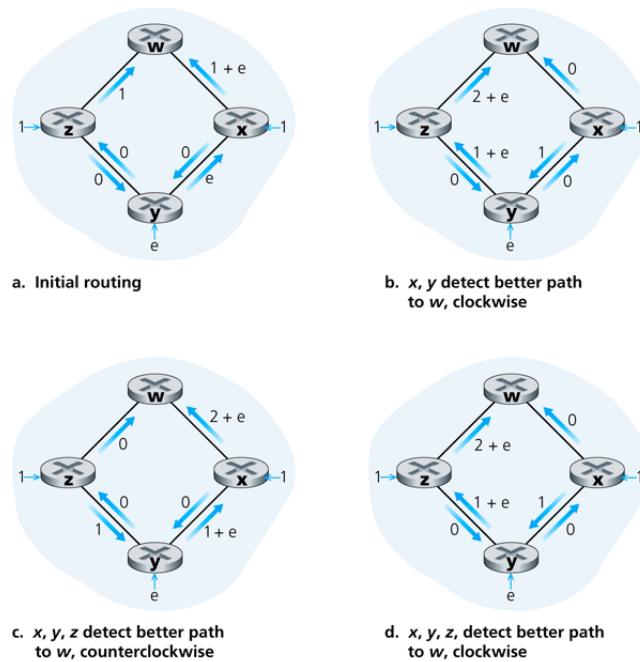


Figure 4.29 ♦ Oscillations with congestion-sensitive routing

### Pregunta 4 (10 puntos)

Dada una red de paquetes conmutados, mencione y describa los tipos de retardo.

#### Solución

Las fuentes de retardo son 4:

##### Procesamiento en el nodo

Por ejemplo, debido a chequeo de paridad (CRC) y/o a determinar enlace de salida.

##### Encolamiento

Debido a espera en colas del enlace de salida para transmisión; depende del nivel de congestión del router.

##### Retardo de transmisión

Si  $R$ =ancho de banda del enlace (bps) y  $L$ =longitud del paquete (bits), tiempo de envío =  $L/R$ .

##### Retardo de propagación

Sea  $d$  = longitud del enlace físico y  $s$  = velocidad de propagación en el medio (por ejemplo  $\sim 2 \times 10^8$  m/seg en cobre), entonces, el retardo de propagación es =  $d/s$ .

### Pregunta 5 (7 puntos)

Describa las características de las arquitecturas de Capa de Aplicación Cliente-Servidor y Peer-to-Peer. Mencione al menos dos ejemplos de cada una.

#### Solución

##### Cliente-Servidor

En estas arquitecturas, típicamente las aplicaciones tienen dos componentes fundamentales: el cliente y el servidor.

Los requerimientos de los clientes (hosts), a los que no se les requiere ser "always-on", son servidos por servidores, estos sí "always-on"

Cliente: típicamente, quien inicia el contacto con el servidor, requiriendo servicios ofrecidos por él.

Servidor: quien provee el servicio demandado por el cliente.

En general, no hay comunicación directa entre clientes.

Ejemplo1 DNS (Servicio de resolución de nombres)

Ejemplo2 Web

##### Peer-to-Peer

En este tipo de arquitecturas hay un uso mínimo (o nulo) de servidores dedicados.

Todos los peers oficián de clientes y también servidores, lo que le otorga a este tipo de arquitectura la propiedad de ser altamente escalable.

## Redes de Computadoras

Los peers "aparecen" y "desaparecen" de la red, por lo que, desde el rol de servidores, no cumplen con la propiedad de "always-on", en comparación con el caso de la arquitectura Cliente-Servidor.

Estas arquitecturas aprovechan la capacidad de procesamiento y almacenamiento de los peers, así como el ancho de banda de sus conexiones a Internet, liberando de dichos requerimientos a los "servidores tradicionales". Además de diferenciarse de la arquitectura Cliente-Servidor en cuanto al paradigma de funcionamiento, también se diferencia en cómo escalan, ya que en este caso la capacidad de crecimiento está autocontenida en los peers que participan de la misma. En el caso de la arquitectura Cliente-Servidor, la demanda de crecimiento se concentra del lado del Servidor.

Ejemplo1 Bit Torrent (transferencias de archivos).

Ejemplo2 Skype (videollamadas, chat).

## Problemas Prácticos

### Problema 1 (30 puntos)

Sea *ceroconf* un protocolo de descubrimiento de servicios. En este protocolo un servicio se anuncia en la red local enviando mensajes UDP broadcast al puerto 1234. Estos mensajes tienen el formato "nombre\_de\_servicio\número\_de\_puerto", y deben ser enviados cada 30 segundos. Se asume que el servicio es un servidor TCP escuchando en el puerto indicado.

Sea *tcp\_cam* un servicio que permite acceder a una cámara web mediante TCP. Para esto el servicio acepta conexiones TCP y hace streaming en cada conexión leyendo los datos a enviar desde el archivo `/dev/video0`. Este servicio se anuncia a sí mismo usando *ceroconf* con el nombre\_de\_servicio "tcp\_cam".

Sea *video\_capture* una aplicación que espera la aparición de servicios *tcp\_cam* en la red local (anunciados con *ceroconf*). Cada vez que detecta una nueva instancia se conecta a ella y queda leyendo el stream, escribiéndolo a disco en un archivo. La aplicación puede descargar varios streams simultáneamente.

Implemente el servicio *tcp\_cam* y la aplicación *video\_capture* en un lenguaje de alto nivel. Tiene disponible la APIs de sockets, concurrencia y lectura/escritura de archivos, así como bibliotecas de estructuras de datos.

### Solución

```
tcp_cam
=====

service_port = 2222          #donde aceptamos conexiones para streaming
video_file = io.open('/dev/video0', 'r')

function announcer()
    announcement = "tcp_cam\n" + service_port
    u = socket.udp()
    while true do
        u:send('255.255.255.255', 1234, announcement)
        thread.sleep(30)
    end
end

function streamer(fd)
    while not fd:is_closed()
        buff = video_file:read()
        fd:write(buff)
    end
end

master = socket.tcp()
master:bind('localhost', service_port)
thread.new(announcer)
while true do
    fd = master:accept()
    thread.new(streamer, fd)
end
```

```

video_capture
=====

cameras = {} #tabla para llevar la cuenta de conexiones abiertas

u = socket.udp()
u:setsockopt('**', 1234)

function get_stream (ip, port)
    s = socket.tcp(ip, port)
    f = io.open("cam_" + ip + "_" + port + "_" + get_time(), 'rw' ) #nombre de archivo unico
    while not s:is_closed() do
        buff = s:read()
        if buff then f:write(buff) end
    end
    f:close()
    cameras[ip + ":" + port] = false
end

while true do
    ann, source_ip = u:receive() #tomamos nota del ip origen del paquete
    service_name, service_port = split_lines(ann)
    if service_name == "tcp_cam" then
        if not cameras[source_ip + ":" + service_port] then # ¿no lo estamos atendiendo?
            cameras[source_ip + ":" + service_port] = true
            thread.new(get_stream, source_ip, service_port)
        end
    end
end
end
    
```

**Problema 2 (30 puntos)**

Considere la red de la figura, compuesta por dos Sistemas Autónomos (AS 1 y AS 2) y sin conexión a Internet.

Toda la red del AS 2 ya se encuentra numerada con el prefijo 20.20.20.0/24.

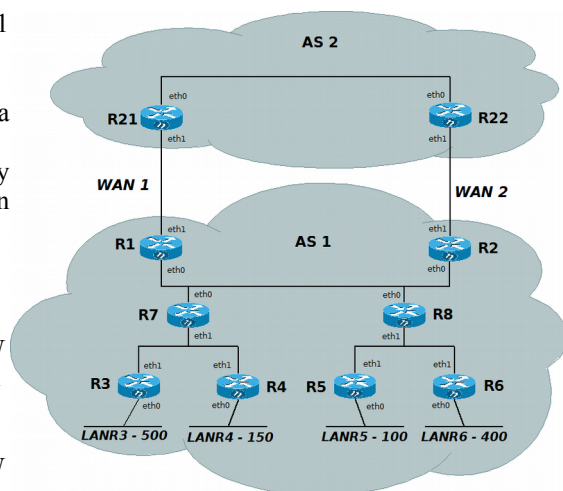
Los enlaces entre ambos Sistemas Autónomos ya se encuentran numerados a partir del prefijo 30.30.30.0/24.

Para cada uno de los segmentos de red LANR3, LANR4, LANR5 y LANR6, se indica la cantidad de computadoras que se encuentran conectadas al mismo.

Los administradores del AS 1 requieren que:

I. el enlace WAN 1 sea la primera opción para el tráfico en ambos sentidos entre LANR3 y LANR4, y la red del AS 2. El enlace WAN 2 debe ser el camino alternativo para dicho tráfico en caso de caída del enlace WAN 1.

- II. el enlace WAN 2 sea la primera opción para el tráfico en ambos sentidos entre LANR5 y LANR6, y la red del AS 2. El enlace WAN 1 debe ser el camino alternativo para dicho tráfico en caso de caída del enlace WAN 2.



Se pide:

- a) Realice el plan de numeración del AS 1 minimizando la cantidad de direcciones IP utilizadas y la cantidad de entradas en las tablas de forwarding de los routers. Dispone para ello de los prefijos que se detallan en la siguiente tabla:

Prefijos	Segmentos LAN
10.10.4.0/22	LANR3 y LANR4
10.10.8.0/22	LANR5 y LANR6
172.16.0.0/24	Entre los routers

- b) Suponiendo que se ejecuta algún protocolo de routing interno, y que se cumplen los requerimientos I y II:
  - i. Escriba las tablas de forwarding de los routers R7 y R2 válidas mientras no hay fallas en ningún enlace.
  - ii. ¿Cómo cambian las tablas de forwarding de los routers R7 y R2 cuando cae el enlace WAN 1?
- c) En la configuración de BGP:
  - i. Indique cómo cumplir los requerimientos I y II solamente con configuraciones locales del AS 1.
  - ii. Indique los prefijos y los atributos AS-PATH que publica cada router en sus sesiones eBGP.



