Examen Parcial II de Sistemas Telemáticos para Medios Audiovisuales

GSyC, Universidad Rey Juan Carlos

13 de enero de 2017

CALIDAD DE SERVICIO y DiffServ

ATENCIÓN:

- **Si ya has usado NetGUI con otro diagrama de red, cierra NetGUI y ejecuta** `clean-netgui.sh` **antes de volver a lanzar NetGUI.**
- **Arranca NetGUI, en el menú "Archivo" elige la opción "Abrir" y escribe como nombre de archivo** `/opt/stma/cs`
- **Se cargará el escenario mostrado en la figura 1.**
- **Arranca cada una de las máquinas del escenario, de una en una.**
- **Si en algún momento quieres volver a tener el escenario en su estado inicial, cierra NetGUI, ejecuta** `clean-netgui.sh` **y ejecuta después** `/opt/stma/cs/reset-lab`

En el escenario no se ha configurado ninguna disciplina de colas ni de entrada ni de salida. Si realizas alguna configuración para alguna de las siguientes preguntas, recuerda borrar dicha configuración antes de pasar a otra pregunta.

---

1. Partiendo de la situación inicial, supongamos que `r2` está limitando el tráfico de entrada en su interfaz `eth0`. Se genera el siguiente tráfico simultáneo durante 10 segundos hacia `pc3` utilizando `iperf`: de `pc2` 1Mbps y de `pc5` 1Mbps.

   Se realiza una captura de tráfico en la interfaz `r2(eth1)` y se obtiene:

   Indica cuál de las siguientes configuraciones en la disciplina de cola en `r2` permite obtener la gráfica previa:

(A)

```
tc filter add dev eth0 parent ffff: protocol ip prio 1 u32 match ip src 13.0.0.20/32 \
    police rate 100kbit burst 5k continue flowid :1

tc filter add dev eth0 parent ffff: protocol ip prio 2 u32 match ip src 13.0.0.20/32 \
   police rate 200kbit burst 5k drop flowid :2

tc filter add dev eth0 parent ffff: protocol ip prio 3 u32 match ip src 13.0.0.50/32 \
    police rate 500kbit burst 5k drop flowid :3
```

(B)

```
tc filter add dev eth0 parent ffff: protocol ip prio 1 u32 match ip src 13.0.0.20/32 \
    police rate 500kbit burst 5k drop flowid :1

tc filter add dev eth0 parent ffff: protocol ip prio 3 u32 match ip src 13.0.0.50/32 \
    police rate 300kbit burst 5k drop flowid :2
```

(C)

```
tc filter add dev eth0 parent ffff: protocol ip prio 1 u32 match ip src 13.0.0.20/32 \
    police rate 200kbit burst 5k continue flowid :1

tc filter add dev eth0 parent ffff: protocol ip prio 2 u32 match ip src 13.0.0.20/32 \
   police rate 100kbit burst 5k continue flowid :2

tc filter add dev eth0 parent ffff: protocol ip prio 3 u32 match ip src 13.0.0.50/32 \
    police rate 500kbit burst 5k continue flowid :3
```

(D)

```
tc filter add dev eth0 parent ffff: protocol ip prio 1 u32 match ip src 13.0.0.20/32 \
    police rate 300kbit burst 5k continue flowid :1

tc filter add dev eth0 parent ffff: protocol ip prio 3 u32 match ip src 13.0.0.50/32 \
    police rate 500kbit burst 5k continue flowid :2
```

2. Partiendo de la situación inicial del escenario se sabe que hay una limitación de 500kbps en el enlace de `r3(eth2)`. Se envía tráfico simultáneo a `pc3` durante 10 segundos desde `pc1`, `pc2` y `pc5` utilizando `iperf` y se ha obtenido la captura de tráfico que se encuentra en el fichero `cap1.cap`.

   Indica cuál de las siguientes afirmaciones es correcta para que dicha captura se haya podido realizar:

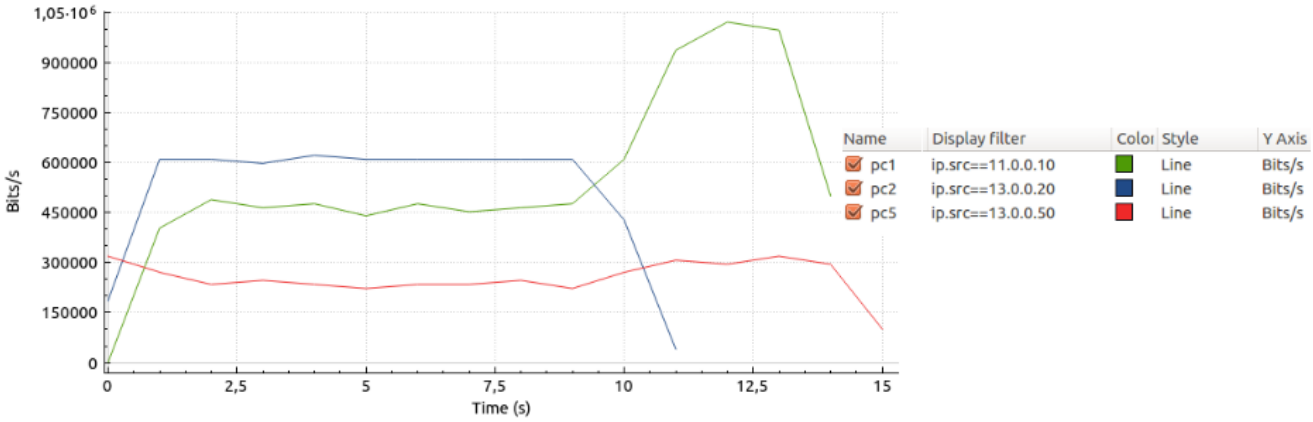
   (A) En `r3(eth2)` se ha configurado una disciplina de cola con prioridad con el siguiente orden de mayor a menor prioridad: `pc1`, `pc2`, `pc5`.

   (B) En `r3(eth2)` se ha configurado una disciplina de cola con prioridad con el siguiente orden de mayor a menor prioridad: `pc2`, `pc1`, `pc5`.

   (C) En `r3(eth2)` se ha configurado una disciplina de cola HTB con el parámetro rate=500kbit y las siguientes limitaciones de tráfico:
   - `pc1`: rate=300kbit ceil=500kbit
   - `pc2`: rate=100kbit ceil=100kbit
   - `pc5`: rate=100kbit ceil=500kbit

   (D) En `r3(eth2)` se ha configurado una disciplina de cola HTB con el parámetro rate=500kbit y las siguientes limitaciones de tráfico:
   - `pc1`: rate=200kbit ceil=500kbit
   - `pc2`: rate=100kbit ceil=100kbit
   - `pc5`: rate=200kbit ceil=500kbit

3. Partiendo de la situación inicial del escenario se configura en `r3` una disciplina de cola, se sabe que `pc1`, `pc2` y `pc5` han estado enviando simultáneamente durante 10 segundos tráfico hacia `pc3` y se ha obtenido la siguiente captura de tráfico:



Indica cuál de las siguientes configuraciones en `r3(eth2)` permitiría haber obtenido dicha gráfica:

(A) Se ha configurado una disciplina de cola HTB con el parámetro rate=1.3Mbit y las siguientes limitaciones de tráfico para las clases hijas:

- `pc1`: rate=400kbit ceil=1.3Mbit
- `pc2`: rate=600kbit ceil=1.3Mbit
- `pc5`: rate=200kbit ceil=300kbit

(B) Se ha configurado una disciplina de cola HTB con el parámetro rate=1Mbit y las siguientes limitaciones de tráfico para las clases hijas:

- `pc1`: rate=400kbit ceil=1Mbit
- `pc2`: rate=600kbit ceil=1Mbit
- `pc5`: rate=200kbit ceil=300kbit

(C) Se ha configurado una disciplina de cola HTB con el parámetro rate=1.3Mbit y las siguientes limitaciones de tráfico para las clases hijas:

- `pc1`: rate=450kbit ceil=1Mbit
- `pc2`: rate=350kbit ceil=1Mbit
- `pc5`: rate=400kbit ceil=400kbit

(D) Se ha configurado una disciplina de cola HTB con el parámetro rate=1Mbit y las siguientes limitaciones de tráfico para las clases hijas:

- `pc1`: rate=450kbit ceil=1Mbit
- `pc2`: rate=350kbit ceil=1Mbit
- `pc5`: rate=400kbit ceil=400kbit

4. Partiendo de la situación inicial del escenario se configura en `r3` la siguiente disciplina de cola de tráfico:

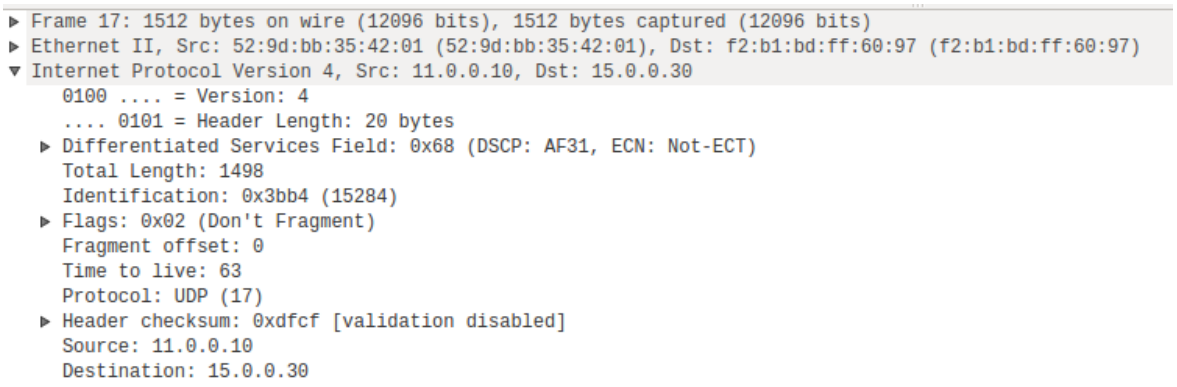
```
tc qdisc add dev eth2 handle 1:0 root dsmark indices 8 set_tc_index
tc filter add dev eth2 parent 1:0 protocol ip prio 1 tcindex mask 0xfc shift 2

tc qdisc add dev eth2 parent 1:0 handle 2:0 htb

tc class add dev eth2 parent 2:0 classid 2:1 htb rate 1Mbit
tc class add dev eth2 parent 2:1 classid 2:10 htb rate 200kbit ceil 1Mbit
tc class add dev eth2 parent 2:1 classid 2:20 htb rate 300kbit ceil 1Mbit
tc class add dev eth2 parent 2:1 classid 2:30 htb rate 500kbit ceil 1Mbit

tc filter add dev eth2 parent 2:0 protocol ip prio 1 handle 0x68 tcindex classid 2:10
tc filter add dev eth2 parent 2:0 protocol ip prio 2 handle 0x3bb4 tcindex classid 2:20
tc filter add dev eth2 parent 2:0 protocol ip prio 3 handle 0x1a tcindex classid 2:30
```

`r3` recibe el siguiente paquete:



Indica cuál de las siguientes afirmaciones es correcta:

(A) El paquete no se clasifica en ninguna de las clases definidas en `r3`.

(B) El paquete se clasifica en la clase 2:10

(C) El paquete se clasifica en la clase 2:20

(D) El paquete se clasifica en la clase 2:30

HTTP

5. Analiza la captura `/opt/stma/http-1.cap`. Suponiendo que la máquina 13.0.0.13 no tuviera almacenada ninguna *cookie* antes del tráfico que aparece en la captura, y que después no hay ningún tráfico hasta el día de hoy, indica qué *cookies* enviará dicha máquina si hoy consulta la URL: `http://elcortebritanico.com/facturas/index.html`

   (A) El cliente envíará exclusivamente las *cookies* `Nombre`, `Nif`, `Edad`, `Carrito`, `Sesion`.

   (B) El cliente envíará exclusivamente las *cookies* `Nombre`, `Nif`, `Edad`, `Sesion`.

   (C) El cliente envíará exclusivamente las *cookies* `Nombre`, `Nif`, `Edad`.

   (D) El cliente no enviará ninguna *cookie*

6. Un cliente HTTP envía la siguiente petición a un servidor HTTP:

```
GET /page_1.html HTTP/1.1
Host: www.server_one.com
Connection: close
```

   Se sabe que la página pedida contiene 3 imágenes, 1 que está en el mismo servidor, y 2 que están en el servidor `www.server_two.com`, con quien el cliente se comunicará usando también HTTP/1.1 y usando la misma cabecera `Connection` que con `www.server_one.com`.

   A partir del momento en el que el cliente termine de recibir completamente el recurso `/page_1.html`, indica cuántas nuevas conexiones TCP abrirá dicho cliente:

   (A) 1.

   (B) 2.

   (C) 3.

   (D) El resto de afirmaciones son falsas.

7. Analiza la captura `/opt/stma/http-2.cap`. Indica cuál de las siguientes afirmaciones es correcta respecto a lo que se observa en los paquetes 3 a 22 de dicha captura.

   (A)
   - La máquina 23.0.0.23 es un servidor proxy de HTTP.
   - El recurso `/index.html` está presente en la caché del proxy, pero ha caducado.
   - El servidor `www2` NO le envía el recurso `/index.html` al proxy.

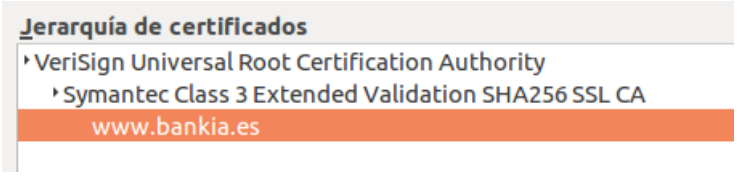
   (B)
   - La máquina 23.0.0.23 es un servidor proxy de HTTP.
   - El recurso `/index.html` NO está presente en la caché del proxy.
   - El proxy SÍ le envía el recurso `/index.html` al cliente.

   (C)
   - La máquina 23.0.0.23 es un servidor proxy de HTTP.
   - El servidor `www2` SÍ le envía el recurso `/index.html` al proxy.
   - El proxy SÍ le envía el recurso `/index.html` al cliente.

   (D)
   - La máquina `www2` es un servidor proxy de HTTP.
   - El recurso `/index.html` está presente en la caché del proxy, y está vigente (es decir, aún no ha caducado).
   - El servidor proxy NO le envía el recurso `/index.html` al cliente, porque la versión que ya tiene no ha sido modificada.

SEGURIDAD

8. Al acceder a la página web `www.bankia.es` desde el navegador, se observa que la información que muestra la página sobre la jerarquía de certificados es la siguiente:



Indica cuál de las siguientes afirmaciones es correcta:

(A) El certificado de www.bankia.es está instalado en el navegador por ser certificado de una autoridad de certificación raíz y está autofirmado.

(B) El certificado de www.bankia.es está instalado en el navegador por ser certificado de una autoridad de certificación raíz y estará firmado por VeriSign que es otra autoridad de certificación de confianza

(C) El certificado de Verisign está instalado en el navegador por ser certificado de una autoridad de certificación raíz y este certificado estará firmado por otra autoridad de certificación de confianza.

(D) El certificado de Verisign está instalado en el navegador por ser certificado de una autoridad de certificación raíz y este certificado estará autofirmado.

9. En una sistema existe la autoridad de certificación raíz CA1 que ha incluido su propio certificado autofirmado en la aplicación de comunicaciones que se usa dentro de este sistema. Alicia tiene un certificado de su clave pública firmado por CA1. Roberto no tiene ningún certificado de su clave pública.

Indica cuál de las siguientes afirmaciones es correcta:

(A) Cuando Alicia le envía el certificado de su clave pública a Roberto, Roberto puede estar seguro de que la K_A^+ incluida en el certificado pertenece a Alicia.

(B) Aunque Roberto no reciba el certificado de la clave pública de Alicia, Roberto puede estar seguro de que la K_A^+ pertenece a Alicia ya que Alicia ha tenido que acreditar su identidad ante la autoridad de certificación CA1.

(C) Gracias al certificado de clave pública de Alicia, Alicia puede enviar mensajes confidenciales a Roberto.

(D) El certificado de clave pública de Alicia permite que Alicia pueda verificar usando K_A^+ la integridad de los mensajes que le envía Roberto.

10. Alicia y Roberto usan criptografía de clave pública para intercambiar mensajes. Se sabe que Alicia y Roberto se han intercambiado sus claves públicas de forma segura.

Alicia decide que a partir de ahora van a usar criptografía de clave simétrica y para ello Alicia eligirá una K_s y se la comunicará a Roberto.

Alicia y Roberto usarán K_s para mantener conversaciones con las siguientes propiedades: confidencialidad, autenticidad e integridad. Indica cuál de los siguientes mensajes es la mejor forma para que Alicia envíe a Roberto la clave K_s.

(A) $\boxed{K_A^+(K_s),\ \ H(K_s)}$

(B) $\boxed{K_A^-(K_s),\ \ H(K_R^+(K_s))}$

(C) $\boxed{K_R^+(K_s),\ \ K_A^-(H(K_R^+(K_s)))}$

(D) El resto de afirmaciones no son adecuadadas para enviar K_s y garantizar su uso con las propiedades mencionadas previamente.

11. Alicia, Roberto y Bárbara usan criptografía de clave pública para intercambiar mensajes. Se sabe que:

    - Alicia y Roberto se han intercambiado sus claves públicas de forma segura.
    - Alicia y Bárbara se han intercambiado sus claves públicas de forma segura.

    Roberto necesita la clave pública de Bárbara pero no la tiene.

    Indica cuál de los siguientes mensajes es la mejor forma para que Alicia pueda enviar la clave pública de Bárbara a Roberto.

    (A) $\boxed{K_B^+,\ K_A^-(H(K_B^+))}$

    (B) $\boxed{K_R^+(K_B^+)}$

    (C) $\boxed{K_B^+,\ K_R^+(H(K_B^+))}$

    (D) $\boxed{K_R^+(K_B^+),\ K_R^+(H(K_B^+))}$

IPTABLES

ATENCIÓN:

- **Si ya has usado NetGUI con otro diagrama de red, cierra NetGUI y ejecuta** `clean-netgui.sh` **antes de volver a lanzar NetGUI.**
- **En NetGUI, en el menú "Archivo" elige la opción "Abrir" y carga el nombre de archivo** `/opt/stma/seg`**.**
- **Se cargará el escenario mostrado en la figura 2.**
- **NO ARRANQUES NINGUNA MÁQUINA. Es importante que las arranques en el orden indicado.**
- **Si en algún momento quieres volver a tener el escenario en su estado inicial, cierra NetGUI, ejecuta** `clean-netgui.sh` **y ejecuta después** `/opt/stma/seg/reset-lab`**.**

En la figura 2 se muestra la conexión de dos pequeñas empresas a Internet a través de un proveedor de servicios de Internet (ISP). Estas entidades quedan representadas en la figura de la siguiente forma:

- Empresa1: tiene las siguientes máquinas `e1-pc1` y `e1-pc2` que pertenecen a una subred privada, `e1-pc3` y `e1-pc4` que pertenecen a una zona DMZ y el *router firewall* `e1-fw`.
- Empresa2: tiene las siguientes máquinas `e2-pc1`, `e2-pc2` que pertenecen a una subred privada y el *router firewall* `e2-fw`.
- ISP: tiene un único *router* `isp-r1`.
- Internet: tiene las siguientes máquinas `i-pc1`, `i-pc2` y los siguientes *routers* `i-r1` y `i-r2`.

Las máquinas `e1-fw` y `e2-fw` están funcionando como *firewalls* a los que se les ha configurado únicamente las siguientes reglas:

- Políticas por defecto para las cadenas de entrada y reenvío (`INPUT` y `FORWARD`) configuradas para **descartar** paquetes.
- Política por defecto para la cadena de salida (`OUPUT`) configurada para **aceptar** paquetes.

Al arrancar el *router* `e1-fw` ha ejecutado el *script* `/bin/fw1.sh` y al arrancar el *router* `e2-fw` ha ejecutado el *script* `/bin/fw2.sh`. Estos *scripts* aplican las reglas descritas previamente.

---

12. Partiendo de la configuración inicial descrita del escenario, se ha aplicado en `e1-fw` la siguiente configuración:

```
iptables -t filter -A FORWARD -s 10.0.0.10 -p tcp -j ACCEPT
iptables -t filter -A FORWARD -s 10.0.0.20 -p tcp --dport 1000 -j ACCEPT
iptables -t filter -A FORWARD -i eth0 -o eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

A continuación se ejecuta en `e1-pc1`:

```
e1-pc1:~# nc -l -p 1000
```

Y en `e1-pc2`:

```
e1-pc2:~# nc -l -p 1000
```

Indica cuál de las siguientes afirmaciones es correcta:

(A) Desde `i-pc1` SÍ podrá comunicarse un cliente con el servidor lanzado en `e1-pc1`, pero NO podrá comunicarse con el servidor lanzado en `e1-pc2`.

(B) Desde `i-pc1` NO podrá comunicarse un cliente con el servidor lanzado en `e1-pc1`, pero SÍ podrá comunicarse con el servidor lanzado en `e1-pc2`.

(C) Desde `i-pc1` SÍ podrá comunicarse un cliente con el servidor lanzado en `e1-pc1`, y TAMBIÉN podrá comunicarse con el servidor lanzado en `e1-pc2`.

(D) Desde `i-pc1` NO podrá comunicarse un cliente con el servidor lanzado en `e1-pc1`, ni TAMPOCO podrá comunicarse con el servidor lanzado en `e1-pc2`.

13. Partiendo de la configuración inicial, se añaden a la configuración de `e1-fw` las siguientes reglas:

```
[REGLA 1]     iptables -t filter -A INPUT -s 20.0.0.40 -j ACCEPT
[REGLA 2]     iptables -t filter -A OUTPUT -p icmp -j DROP
[REGLA 3]     iptables -t filter -A OUTPUT -d 20.0.0.40 -j ACCEPT
```

Indica cuál de las siguientes afirmaciones es correcta:

(A) Con esa configuración NO funcionará un *ping* entre `e1-pc4` y `e1-fw` ya que la política DROP de la cadena INPUT se aplica antes que la REGLA 1.

(B) Con esa configuración SÍ funcionará un *ping* entre `e1-pc4` y `e1-fw` gracias a la REGLA 1 y a que la REGLA 3, que al ser más específica que la REGLA 2 se aplica antes.

(C) Con esa configuración SÍ funcionará un *ping* entre `e1-pc4` y `e1-fw` ya que la política ACCEPT de la cadena OUTPUT se aplica antes que la REGLA 2 y que la REGLA 3.

(D) El resto de afirmaciones son falsas.

14. En la máquina `i-pc1` se arranca un servidor TCP esperando recibir mensajes en el puerto 9000.

    Partiendo de la configuración inicial, indica cuál de los siguientes conjuntos de reglas adicionales en `e1-fw` permite que un cliente TCP lanzado en cualquier máquina de la Empresa1 se comunique con dicho servidor:

    (A)
    ```
    iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -o eth0 -j SNAT --to-source 20.0.1.1
    iptables -t filter -A FORWARD -o eth0 -p tcp --dport 9000 -j ACCEPT
    iptables -t filter -A FORWARD -i eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
    ```

    (B)
    ```
    iptables -t nat -A PREROUTING -d 20.0.4.10 -o eth0 -p tcp --dport 9000 -j DNAT --to-source 20.0.1.1
    iptables -t filter -A FORWARD -o eth0 -p tcp --dport 9000 -j ACCEPT
    iptables -t filter -A FORWARD -i eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
    ```

    (C)
    ```
    iptables -t nat -A PREROUTING -d 20.0.4.10 -o eth0 -p tcp --dport 9000 -j DNAT --to-source 20.0.1.1
    iptables -t filter -A FORWARD -o eth0 -i eth1 -p tcp --dport 9000 -j ACCEPT
    iptables -t filter -A FORWARD -o eth0 -i eth2 -p tcp --dport 9000 -j ACCEPT
    iptables -t filter -A FORWARD -i eth0 -o eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT
    iptables -t filter -A FORWARD -i eth0 -o eth2 -m state --state RELATED,ESTABLISHED -j ACCEPT
    ```

    (D)
    ```
    iptables -t filter -A FORWARD -o eth0 -i eth1 -p tcp --dport 9000 -j ACCEPT
    iptables -t filter -A FORWARD -o eth0 -i eth2 -p tcp --dport 9000 -j ACCEPT
    iptables -t filter -A FORWARD -i eth0 -o eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT
    iptables -t filter -A FORWARD -i eth0 -o eth2 -m state --state RELATED,ESTABLISHED -j ACCEPT
    ```

15. Partiendo de la configuración inicial, indica cuál de los siguientes conjuntos de reglas aplicadas en `e1-fw` permite que tanto desde las máquinas de Internet como desde todas las máquinas de la Empresa 1 funcione un `ping` a `e1-fw`:

    (A)
    ```
    iptables -t filter -A INPUT -p icmp -j ACCEPT
    ```

    (B)
    ```
    iptables -t filter -A FORWARD -i eth0 -p icmp -j ACCEPT
    iptables -t filter -A FORWARD -i eth1 -p icmp -j ACCEPT
    iptables -t filter -A FORWARD -i eth2 -p icmp -j ACCEPT
    ```

    (C)
    ```
    iptables -t filter -A FORWARD -i eth0 -p icmp -j ACCEPT
    iptables -t filter -A FORWARD -i eth1 -p icmp -j ACCEPT
    iptables -t filter -A FORWARD -i eth2 -p icmp -j ACCEPT
    iptables -t nat -A PREROUTING -p icmp -d 20.0.1.1 -j DNAT --to-destination 10.0.0.0/24
    ```

    (D)
    ```
    iptables -t filter -A FORWARD -i eth0 -p icmp -j ACCEPT
    iptables -t filter -A FORWARD -i eth1 -p icmp -j ACCEPT
    iptables -t filter -A FORWARD -i eth2 -p icmp -j ACCEPT
    iptables -t nat -A PREROUTING -p icmp -s 10.0.0.0/24 -j SNAT --to-source 20.0.1.1
    ```

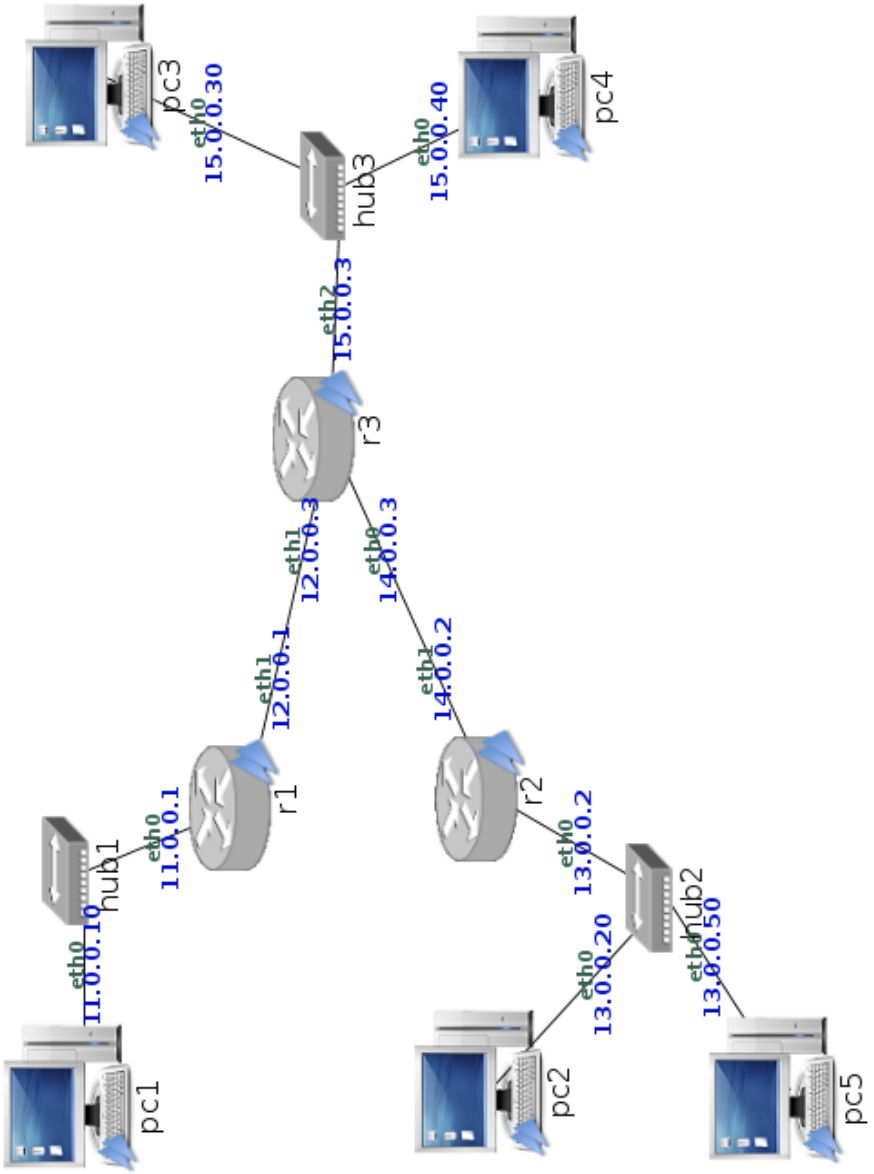


Figura 1: Calidad de servicio

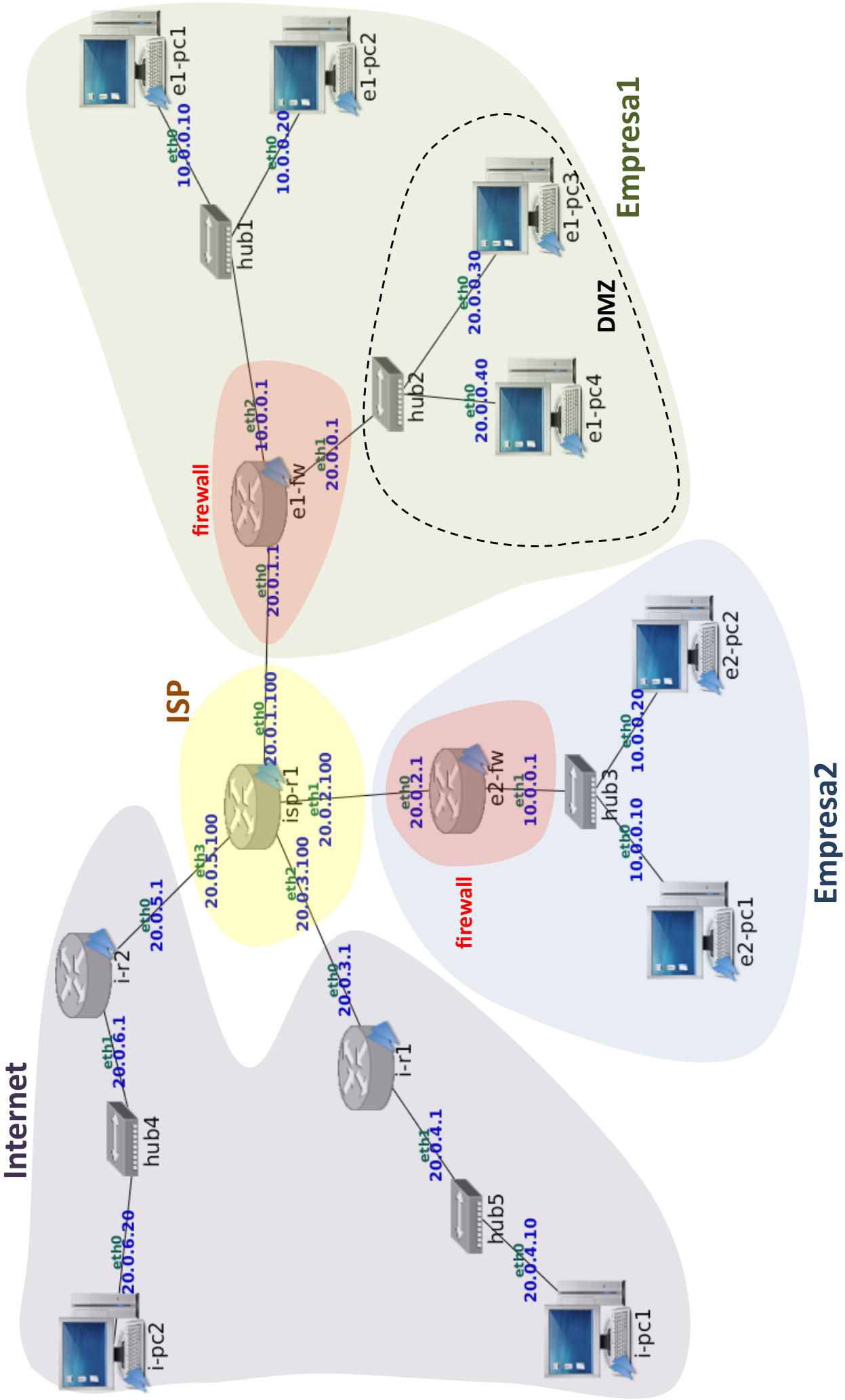


Figura 2: Seguridad