

# Desarrollo de una Pasarela Residencial con restricciones de Calidad de Servicio en accesos de banda ancha

Jaime García, Francisco Valera, Carmen Guerrero, Arturo Azcorra  
Universidad Carlos III de Madrid, Departamento de Ingeniería Telemática,  
Avda. de la Universidad 30, 28911 Leganés (Madrid), España  
Email: {jgr,fvalera,guerrero,azcorra}@it.uc3m.es

## Resumen

*En este artículo se expondrá el trabajo llevado a cabo por los autores dentro del proyecto europeo MUSE, que tiene como misión crear un estándar europeo de redes de banda ancha que cumpla con los objetivos planteados para que llegue a la mayor parte de los ciudadanos de este continente, cumpla sus expectativas de alto rendimiento y sea económica su contratación. Dentro de este proyecto existen varios puntos de interés aunque nosotros veremos el equipo que hace de interfaz entre la red de acceso y la red del usuario denominado Residential Gateway (RGW). Dentro del proyecto MUSE se ha llegado a un acuerdo satisfactorio sobre la arquitectura del RGW por lo que se debe crear un prototipo para comprobar la viabilidad del mismo. En este artículo se describirán los pasos realizados hasta llegar al prototipo final, explicando las fases de elección de software y hardware, los modelos ideados para la implementación, las pruebas realizadas para validar dichos modelos y la implementación final del prototipo.*

## 1. Introducción

La tecnología de acceso existente hoy en día permite llevar a los usuarios finales los servicios de banda ancha que tantas veces se ha venido anunciando en los últimos años. Existen todavía no obstante, otros problemas ajenos como pueden ser los altos costes de implantación, que van disminuyendo rápidamente y existen países donde la fibra óptica ya llega a muchos entornos residenciales. MUSE (Multi Service Access Everywhere, [1]) es un gran proyecto Europeo englobado en el sexto Programa Marco, cuyo objetivo principal es el de investigar y proporcionar desarrollos en las redes de acceso y de agregación del futuro, para permitir a los ciudadanos europeos acceder a servicios reales de banda ancha a bajo coste. En la Fig. 1 se muestra el campo de acción del proyecto MUSE.

Uno de los puntos clave del proyecto MUSE, es la pasarela residencial o Residential Gateway (RGW). Este dispositivo se encuentra ubicado en la frontera entre la red de acceso y la red del usuario y debe soportar por tanto la amplia gama de tecnologías utilizadas a ambos lados como pueden ser: Ethernet, 802.11a/b/g, USB, Firewire, Bluetooth, etc. para las redes del hogar y ATM, xDSL, Ethernet, etc. en las redes de acceso. Además de realizar la conversión

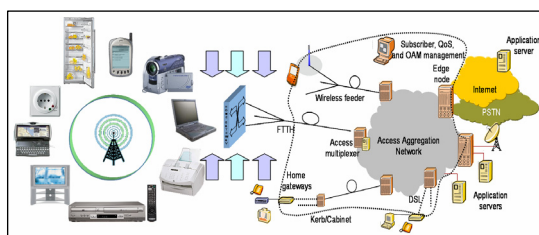


Fig. 1. Acciones en el proyecto MUSE

entre tecnologías, otras funciones importantes de un RGW son: conversión entre protocolos de señalización, reenvío o creación de tramas de señalización, comportamiento en modo bridge o router (servicios a nivel 2 o a nivel 3) dependiendo de las características de los flujos, administración de red y servicios, petición de nuevos servicios, auto configuración, nomadismo, seguridad, capacidad de mejora en línea, calidad de servicio real extremo a extremo, pilas IPv4/IPv6, servicios de DHCP/DNS, etc.

En este artículo describiremos el trabajo realizado dentro del proyecto MUSE a la hora de diseñar e implementar el prototipo de este RGW. Se mostrará la arquitectura propuesta que surge de la colaboración con otros organismos de estandarización y la elección de software y hardware para el prototipo. Con el fin de validar el modelo propuesto, se presentan una serie de pruebas reales donde se realizan estudios exhaustivos que representan un escenario de banda ancha como el que se busca en MUSE.

## 2. Arquitectura del RGW

Dentro del proyecto MUSE se escogió seguir las pautas de otros organismos de estandarización para crear la arquitectura del RGW. En especial, la arquitectura base elegida es la propuesta en el DSL-Forum [2] usando también parte de las del FS-VDSL Forum [3].

Al querer diseñar un RGW flexible que sea fácil de adaptar a cualquier entorno, se eligió separar el RGW en varios bloques funcionales que realizarán misiones muy concretas. Estos bloques funcionales no implican tener diferentes dispositivos siendo

posible implementar varios o incluso todos en un solo equipo.

Los bloques funcionales aceptados son:

- NT1 (Network Termination 1) se encarga de la adaptación de la red de acceso a una red Ethernet que es el estándar en la RGW.
- NT2 (Network Termination 2) es el bloque funcional principal ya que se encarga de aspectos tales como el de clasificar, priorizar, filtrado, encaminamiento, configuración, etc.
- ST3 (Service Termination 3) donde termina la conexión del servicio.

### 3. Diseño del prototipo

En la primera fase del diseño del prototipo se detectaron las necesidades principales que tendrían que satisfacer los equipos tanto en hardware como en software. De esta lista, algunas funcionalidades son imprescindibles y otras deseables. La lista resultante es la siguiente:

- Procesamiento a bajo nivel
- Procesamiento de datagramas IP (v4 y/o v6)
- Disponibilidad de varios interfaces de red
- Mantenimiento, vitalidad y mejoras
- Portabilidad
- Rendimiento
- Seguridad

Con estos requisitos en mente se eligió utilizar un dispositivo compatible con la arquitectura i386 de Intel, ya que en la fase de desarrollo facilita enormemente la elección de software y es una arquitectura muy extendida por lo cual resulta muy fácil satisfacer los requisitos expuestos. En concreto, se eligió un dispositivo Book PC, de pequeñas dimensiones para realizar las demostraciones ya que es lo que visualmente se espera de una RGW.

Linux es el sistema operativo que se utilizará en este prototipo debido a sus grandes ventajas: código abierto y libre, gran vitalidad y mantenimiento, portabilidad, seguridad, pilas IPv4 e IPv6, etc. Si unimos Linux y el ordenador compacto elegido, cumplimos los requisitos expuestos excepto el procesamiento de tramas a bajo nivel. En el núcleo estándar de Linux no existe una herramienta lo suficientemente flexible como para poder trabajar con tramas de nivel de enlace, por lo que se necesitó buscar otro software que complementase a Linux. Después de estudiar varias posibilidades (iptables, ebtables, netlink, libpcap, modificación del núcleo de Linux, etc.) se optó por utilizar el software de creación de routers modulares Click [4].

Click fue desarrollado conjuntamente por el MIT, ICSI y UCLA y se compone de una serie de *elementos* programados que al unirlos con un lenguaje propio se pueden realizar diferentes funcionalidades como pueden ser un firewall, una caja NAPT o un router completo. La gran ventaja de utilizar Click (al usar Click como un módulo del kernel de Linux) es la de tener total libertad para capturar, modificar e insertar tramas a cualquier nivel.

### 4. Arquitectura del prototipo

Una vez elegidas las piezas del prototipo, nos queda la implementación de las funcionalidades descritas en la arquitectura del RGW. Debido a que no existen todos los elementos necesarios para desarrollar la RGW en Click y a que desarrollar esos nuevos elementos en Click implicaría la programación a nivel del kernel de Linux, lo cual disminuiría la portabilidad del código escrito, se decidió utilizar un modelo híbrido donde se implementa a nivel de Click todo lo posible y las nuevas funcionalidades se escriben a nivel de aplicación. En este nuevo modelo aparece entonces una aplicación a nivel de aplicación que llamaremos el CSD (*Click Signalling Dispatcher*) y que se encargará del tratamiento de tramas que no puedan ser tratadas a bajo nivel. Este modelo se puede ver en la Fig. 2 donde se aprecia el funcionamiento del mismo. Los SPx (Signalling Process x) son aplicaciones que se ejecutan a nivel de usuario y que se registran con el CSD al ser invocadas. Es en ese momento cuando el CSD modifica el nivel Click para que le envíe unas determinadas tramas que reenviará posteriormente al SP adecuado. Por ejemplo, si el SP se encarga de manejar señalización SIP, el CSD configurará Click para que le envíe todas las tramas SIP que luego redirigirá al SP. Este modelo puede extenderse a otro tipo de tramas que no sean de señalización.

Con este nuevo modelo en mente, se describe la forma de implementar cada uno de los bloques funcionales descritos en la arquitectura. Como se ha mencionado anteriormente, algunos de estos bloques

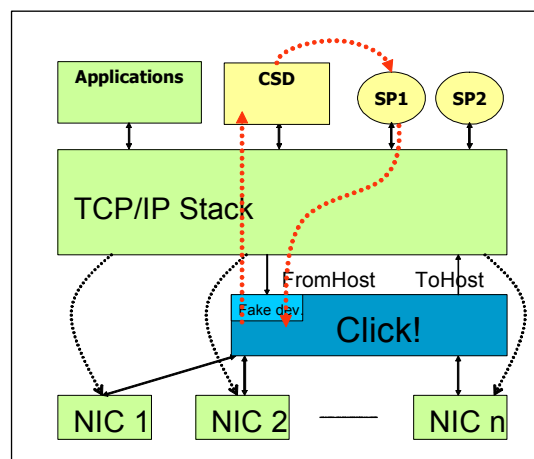


Fig. 2. Modelo híbrido

se realizarán con elementos de Click, mientras que otros se realizarán con una programación a alto nivel (por ejemplo en Java). La Fig. 3 muestra los distintos bloques funcionales que tendrán que implementarse y también el nivel en el que se codificarán. Como se puede ver en dicha figura, existen dos tipos de bloques funcionales a nivel Click, dependiendo del sentido de los paquetes: en el sentido upstream los paquetes van de la red del usuario hacia la red de acceso y en downstream lo hacen en el sentido opuesto. Se realiza esta diferenciación para hacer notar que los elementos Click son independientes en ambos sentidos para garantizar una Calidad de Servicio apropiada. La única excepción es la del NAPT como se explicará posteriormente.

#### 4.1. Definición de los bloques funcionales

A continuación se comentará la función y forma de implementación de los distintos bloques funcionales que juntos realizan las funciones del RGW.

- *Classifier/VLAN Tagging.* En el proyecto MUSE se decidió utilizar el estándar IEEE 802.1Q/p [5] para las redes con acceso Ethernet, por lo que todas las tramas que circulen por la red de acceso deben tener el formato adecuado. Este bloque clasifica las tramas salientes según parámetros configurados en el RGW y según la prioridad asignada, construye la cabecera adecuada marcando los p-bits según dicha prioridad. Para desarrollar este bloque fue necesario la creación de un nuevo elemento Click que insertase la cabecera 802.1Q.

Con este nuevo elemento, la codificación del bloque es trivial.

- *Policing/Traffic Shaping.* En MUSE se considera que cada CoS posee unos parámetros bien definidos. Entre ellos está el throughput asignado para cada CoS, por lo que el RGW debe aplicar los oportunos algoritmos de policing y/o traffic shaping para asegurar el correcto funcionamiento del mismo. En Click existen elementos que permiten implementar estas políticas por lo que su implementación resultó muy fácil.
- *Dispatcher.* Este bloque funcional tan sólo debe insertar las tramas en la cola adecuada, basándose en el campo de los p-bits de la trama. En Click existen elementos clasificadores que permiten implementar este bloque.
- *Queues.* En cada dirección, existirán tantas colas como CoS definidas. Aunque MUSE no impone un número exacto de CoS, la mayoría de los documentos señalan que el número óptimo de CoS es de cuatro, por lo que este es el número utilizado en este prototipo sin pérdida de generalidad ya que se prevé poder configurar este parámetro. En Click existen los elementos *queue* que permiten crear colas de un tamaño dado.
- *Scheduling.* El scheduler es el mecanismo utilizado para extraer paquetes de las colas en un orden determinado. Existen diferentes algoritmos que permiten la búsqueda de, dada una determinada ocupación de las colas, extraer el paquete

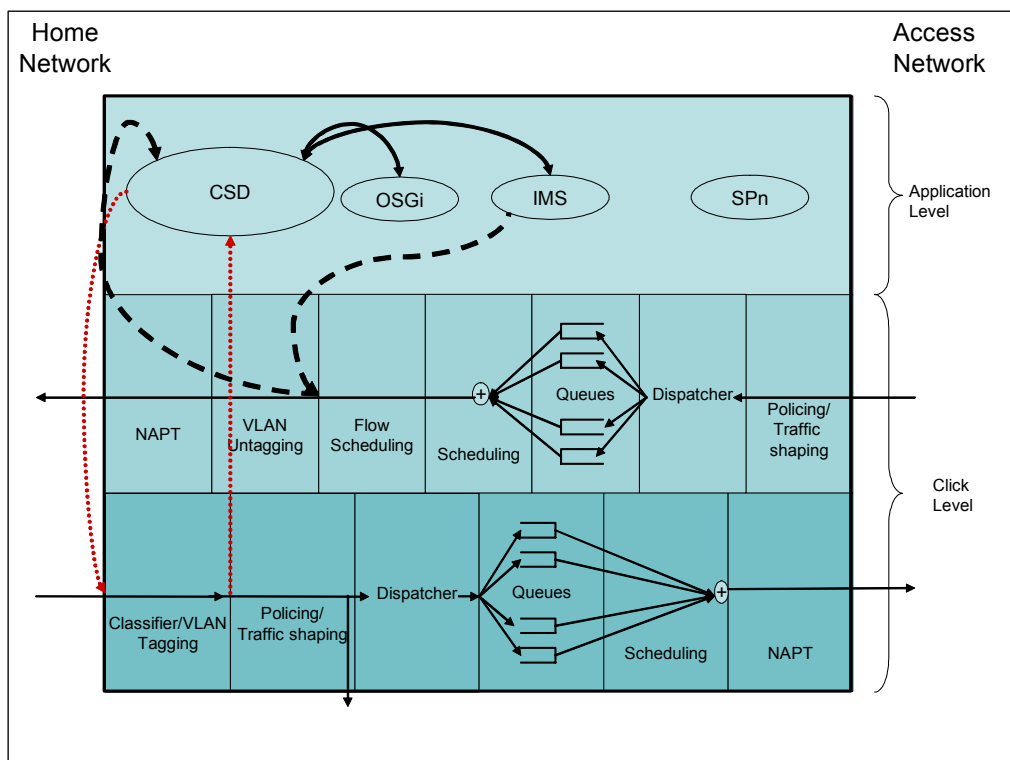


Fig. 3. Bloques funcionales del RGW

de una de ellas que maximice (o minimice) un determinado parámetro, como puede ser el tiempo de retardo de tránsito, el throughput, etc. Este parámetro puede ser idéntico para todos los paquetes en todas las colas, o priorizar cada una de ellas asignándole un valor diferente para cada una de ellas. En Click, existen pocos algoritmos de scheduling disponibles: Round Robin, Deficit Round Robin y Priority Scheduling. Para realizar este prototipo nos hemos decantado por este último elemento, aunque sabemos que no es el algoritmo óptimo para nuestras necesidades y ya estamos trabajando para crear un nuevo elemento que se base en el algoritmo de WFQ (Weighted Fair Queueing) y otro en el CB-WFQ (Class-Based Weighted Fair Queueing) para luego realizar un estudio comparativo.

- *NAPT (Network Address and Port Translation)*. Este bloque resulta imprescindible para el prototipo del RGW ya que en MUSE se ha decidido que los usuarios residenciales usen direcciones privadas en sus equipos y sólo posean una o varias direcciones IP públicas en su interfaz WAN. En Click, existen varios elementos que juntos nos permiten realizar una caja NAPT. Como se ha comentado anteriormente, aunque gráficamente coloquemos dos cajas NAPT diferentes para los sentidos upstream y downstream, estos dos bloques funcionales comparten una tabla en común donde se leen y escriben las correspondencias entre puertos y direcciones IP.
- *Flow Scheduling*. Este es uno de los elementos principales de nuestro modelo híbrido ya que se encarga de extraer o crear copias de tramas para enviarlas hacia el CSD para su posterior tratamiento por un SP.

## 5. Pruebas del modelo híbrido

Antes de implementar los bloques funcionales, existe la necesidad de probar el modelo híbrido propuesto. Los parámetros que se desean probar son el retardo introducido por el modelo híbrido (tramas que deben ser enviadas desde Click a una aplicación a nivel de usuario para ser procesada y luego enviada nuevamente a Click) y también el throughput máximo admisible en este modelo utilizando el ordenador compacto previamente descrito.

Se utilizaron diferentes escenarios de pruebas (modelo híbrido denominado modelo Manager en las pruebas, Click puro y Linux puro) donde se conectaban dos ordenadores (cliente y servidor) al RGW. El RGW realizaba un simple NAT utilizando

diferentes implementaciones según el escenario. En cada uno de estos escenarios se realizaron diferentes pruebas variando la tasa de transmisión desde el cliente y el tamaño de paquetes generados.

La Fig. 4 presenta los resultados de estas pruebas (se han omitido los resultados de las pruebas de Linux puro ya que son similares a las del Click puro) donde se constata que el throughput es un parámetro que depende tanto de la longitud del paquete como de la tasa de transmisión. Es importante notar que en el modelo híbrido las prestaciones dependen fuertemente del tamaño del paquete, y tanto es así, que para paquetes pequeños las prestaciones decaen rápidamente. Debemos recordar que el objetivo principal del modelo híbrido es el de tratar tramas de señalización, lo que será un porcentaje muy bajo del tráfico total. Además, el tamaño mínimo de tramas SIP es de alrededor de los 400 bytes que se corresponden a tramas OK.

Para comprobar el efecto del procesador en las pruebas, se cambió el RGW por otro equipo más potente y se repitieron todas las pruebas. El resultado mostrado en la Fig. 5 demuestra que es importante la potencia del procesador, ya que los resultados mejoran notablemente aunque se sigue observando la dependencia del tamaño del paquete.

De todos estos resultados se desprende que el modelo híbrido es válido siempre y cuando se sigan tratando tramas de datos a nivel Click, y se restrinja el procesamiento de tramas a nivel de aplicación a tramas de señalización, que como se ha dicho, representan un porcentaje muy bajo del total del tráfico que admitirá el RGW y además el tamaño mínimo de trama es lo suficientemente grande como para no representar un problema.

## 6. Implementación del prototipo

Una vez terminadas las fases de diseño y prueba del modelo escogido para el desarrollo del prototipo del RGW, comienza la fase de implementación propiamente dicha. Aunque en las fases previas ya existió un primer contacto con los elementos de

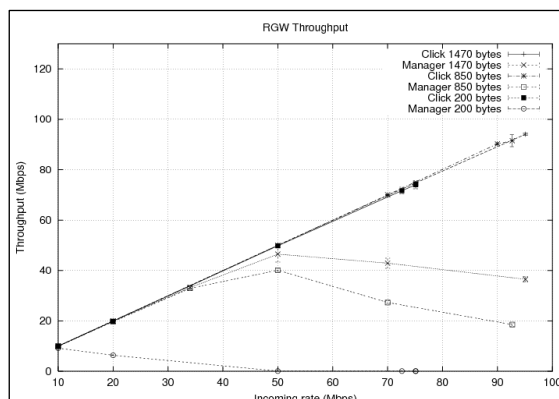
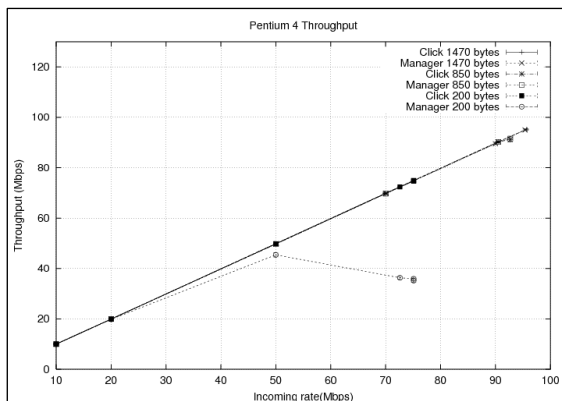


Fig. 4. Throughput utilizando un ordenador compacto



**Fig. 5. Throughput utilizando un Pentium 4**

Click, es ahora cuando se debe tener claro qué elementos son necesarios y cómo se deben interconectar cada uno de ellos para realizar los distintos bloques funcionales. Una de las principales condiciones impuestas a la hora de implementar los bloques fue la de ser totalmente modular; esto es, definir unos interfaces de entrada y salida en los bloques que sean independientes de los bloques que le preceden y/o siguen.

Tan importante como la creación de los bloques funcionales para la implementación del prototipo es la de crear herramientas que faciliten las etapas de pruebas siguientes. Para ello se decidió instalar un *servlet* en el propio RGW para poder reconfigurar el nivel de Click utilizando un interfaz web. La parte central del *servlet* es el programa *FlowTranslator* quien se encarga de, una vez invocado, modificar los flujos definidos por el usuario de una manera intuitiva, a código propio de Click el cual es bastante complejo y poco legible. Una vez generado dicho código Click se instala como la nueva versión del nivel Click del RGW.

En la Fig. 6 se puede ver una captura de una parte del interfaz web presentado al usuario, en concreto los campos para la definición de flujos en el sentido downstream. Un flujo se convertirá en una nueva regla de Click que realizará lo definido en el campo *Action*. Por ejemplo, si el usuario define un flujo donde Dest IP: 192.168.1.1 y Action: Allow as unicast, entonces se creará una nueva regla en Click que dejará pasar todos los datagramas que lleven dicha dirección IP en su campo de dirección destino.

### 6.1. Implementación del nivel Click

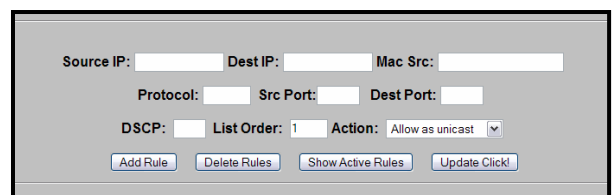
Como se comentó en apartados anteriores, el nivel Click está dividido realmente en dos niveles: downstream y upstream que procesarán tramas en dichos sentidos de transmisión.

#### 6.1.1. Implementación del Upstream

La Fig. 7 representa el orden de conexión de los diferentes elementos de Click para formar el bloque entero de upstream. No se ha dibujado el esquema que representa cuáles son cada uno de los bloques funcionales para no incrementar la complejidad de la figura, pero se puede ver claramente cuáles son cada uno de ellos.

El camino completo de un paquete es el que sigue:

- Los paquetes son extraídos de los diferentes dispositivos de red local.
- Seguidamente los paquetes son pasados a un clasificador. Si se trata de una trama ARP dirigida para el RGW se responde y se vuelve a enviar. Si la trama cumple una de las reglas preconfiguradas (existen reglas globales para todo RGW encargadas de tratar con tramas especiales como pueden ser DHCP, http, ping, etc.) se cumple su acción que generalmente será la de enviar dichas tramas hacia el nivel superior para ser procesadas por el propio RGW. Si no cumple ninguna de estas reglas se pasa al clasificador de reglas definidas por el usuario. Si se cumple alguna de esas reglas se realiza la acción marcada y si no se descarta la trama.
- El siguiente bloque es el encargado de almacenar las tramas en diferentes colas, según lo marcado en las reglas (posibles acciones en los flujos upstream son: Mark as Real Time, Mark as Streaming, Mark as Transactional o Mark as Best Effort). Las salidas de estas colas son conectadas a la entrada del elemento *PrioSched* que es un elemento disponible en Click que implementa un scheduler que usa como algoritmo de sondeo un sistema de prioridades absoluto: si existen paquetes en la cola más prioritaria se extraen elementos de dicha cola hasta vaciarla para seguir luego con la siguiente cola y así hasta terminar en la cola menos prioritaria. Este algoritmo tiene el gran problema de sufrir de inanición por lo que se está estudiando la creación de un nuevo elemento que implemente mejores algoritmos.
- El bloque NAPT es el siguiente paso de los paquetes. Se ha utilizado el elemento *IPRewriter* de Click para implementar dicho bloque. *IPRewriter* es un elemento que sirve para modificar las direcciones IP



**Fig. 6. Definición de flujos downstream**

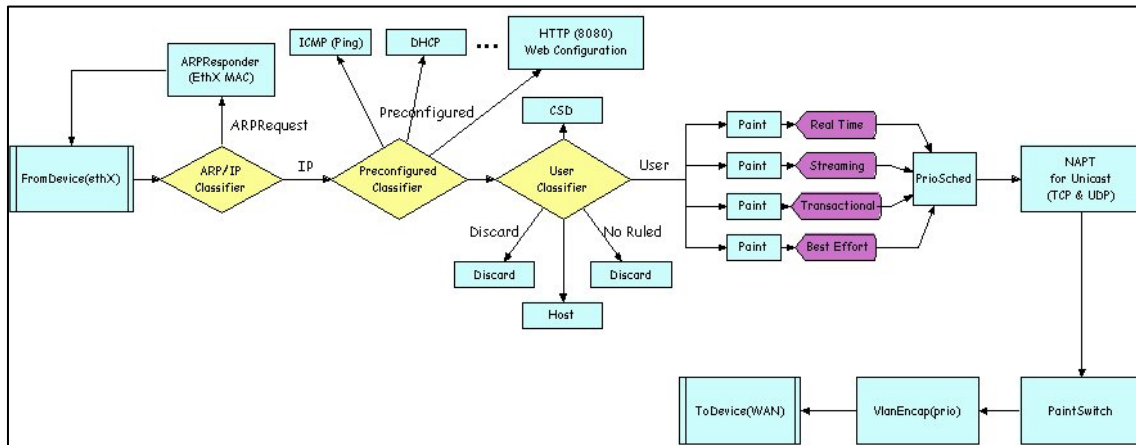


Fig. 7. Interconexión en upstream

y los puertos de los paquetes TCP y UDP (sólo sirve para estos protocolos de transporte). Almacena en una tabla interna los flujos que le llegan y a qué direcciones y qué puertos cambia. Los datos que tiene que cambiar se especifican como parámetro cuando se crea el elemento. En nuestro caso interesa que cambie todos los datos de origen y mantenga los de destino. Como datos de origen siempre pondrá la misma IP origen y asignará a cada conexión un puerto diferente de entre un rango elegido. Este mismo elemento se utiliza tanto para upstream como para downstream, dado que las tramas que lleguen de respuesta son reencaminadas gracias a la tabla interna que contiene. Este elemento tan sólo tiene en cuenta tramas TCP o UDP. Para permitir el flujo de tramas ICMP existe el elemento *ICMPRewriter* que tendría la misma funcionalidad.

- El elemento *PaintSwitch* se utiliza como paso intermedio para saber los p-bits que se deben introducir en la trama Ethernet que luego hará el elemento *VlanEncap*.

### 6.1.2. Implementación del Downstream

La Fig. 8 representa la interconexión de elementos Click creada para llevar a cabo la funcionalidad del nivel Click en el sentido downstream de la comunicación en el RGW.

Como se puede apreciar en la figura, los elementos empleados y la forma de interconectarlos es similar a la utilizada en el sentido upstream, pero con algunas diferencias significativas.

- Las tramas entrantes vienen encapsuladas con el estándar 802.1Q/p por lo que se debe quitar el encapsulado y tener en cuenta los p-bits de las mismas para luego insertarlas en las colas apropiadas.
- Se decidió que el prototipo del RGW soportaría multicast. Esta decisión complica la fase de implementación ya que las tramas multicast no deben pasar por el NAPT. Esto nos lleva a crear nuevas acciones a la hora de definir flujos y permitir de forma separada tramas unicast y tramas multicast. Según esas reglas, una trama unicast irá al bloque de NAPT mientras que una multicast se saltará dicho bloque para ir al

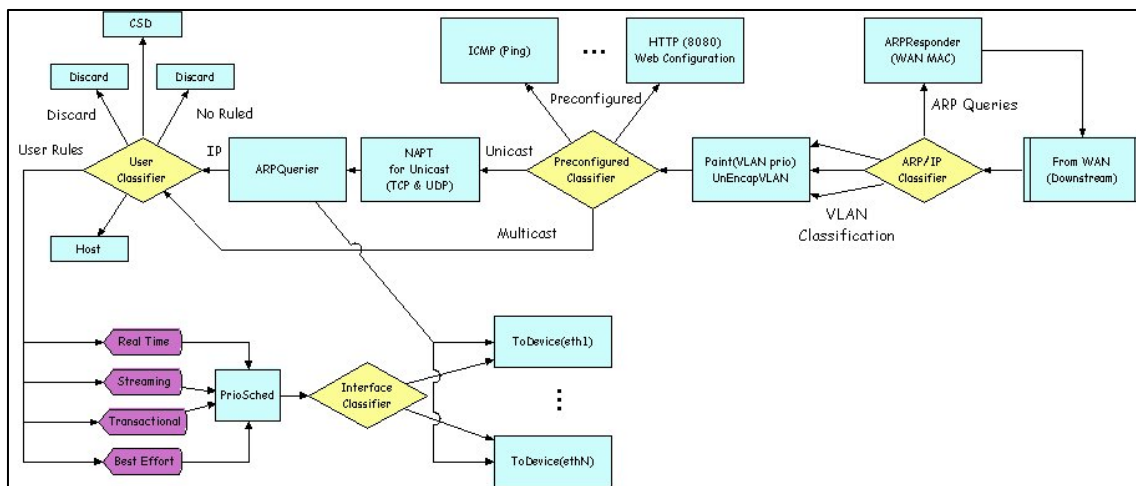


Fig. 8. Interconexión en downstream

resto de reglas de usuario.

- El elemento *IPRewriter* es el único elemento compartido entre los sentidos de upstream y downstream ya que es necesario leer las tablas de NAPT creadas por los flujos salientes.
- A la hora de insertar las tramas en los interfaces de salida se debe detectar de si se trata de una trama unicast o multicast.

El resto de acciones son similares a las del sentido upstream.

## 6.2. Implementación a nivel de aplicación

Son varios los programas creados a nivel de aplicación:

- *CSD*. Como se ha dicho en apartados anteriores, este es el elemento que complementa la funcionalidad del nivel de Click. En nuestro modelo híbrido, el CSD se encarga de realizar el reparto de tramas enviadas desde el nivel de Click hacia los distintos programas en ejecución. Cuando el nivel Click detecta una trama cuya acción es “Send to CSD” este la encapsula en una trama UDP con puerto destino el puerto de escucha del CSD y puerto origen el puerto de escucha de la aplicación SP. El CSD envía la trama al SP correspondiente. Es importante resaltar que un SP al iniciarse se debe registrar primero con el CSD para que este se encargue de crear los nuevos flujos en el nivel Click.
- *Servlet*. Estas aplicaciones ya fueron discutidas al principio de este capítulo.
- *CNC*. El Click Network Controller se encarga de escribir el nuevo fichero de configuración de Click cada vez que existe una modificación del mismo.
- *CBCA (Click Boot Configuration Agent)*. Es el primer programa en ejecutarse. Este programa se encarga de inicializar el resto. Primero debe ejecutar la fase de autenticación (módulo creado por otro participante del proyecto MUSE). Si la autenticación es exitosa debe pasar a ejecutar el cliente de DHCP del que luego se hablará, y luego lanzar el servidor de DHCP para finalmente crear un fichero completo de Click instalándolo como nivel Click.
- *DHCP client*. Este programa creado en Java implementa el protocolo DHCP en un cliente. En una primera aproximación se pensó en usar un cliente DHCP de Linux pero tras varias pruebas se descartó dicha idea.
- *InterfaceConfig*. Este es el programa encargado de encontrar el número de interfaces que posee el RGW para luego

construir el fichero esqueleto el cual se utilizará para generar el fichero Click que se instalará al final de la fase de encendido del RGW.

## 6.3. Funcionalidades extra

Se han realizado algunas funcionalidades extra a las definidas en un principio, sobre todo para realizar de una forma más cómoda la fase de pruebas. Entre ellas destacan:

- *Historial*. Desde el interfaz web se puede elegir guardar la configuración actual. Si el usuario oprime el botón guardar se realiza una copia de los ficheros *current.click* y *config.xml* que son los únicos necesarios para poder realizar una carga posterior de la configuración. Estos ficheros se guardan en un directorio cuyo nombre contiene la fecha y hora del momento de la grabación. El usuario también tiene la opción de poder ver los ficheros guardados con anterioridad, seleccionarlos y volver a un estado previo.
- *Estadísticas de uso*. Otra de las páginas web accesibles en el RGW es la que muestra los principales valores de paquetes y bytes que han pasado por los distintos bloques funcionales. Tanto en downstream como en upstream se tienen valores como por ejemplo el de paquetes/bytes de Real Time, de Streaming, Transactional, Best Effort, TCP, UDP, paquetes descartados porque no habían reglas para ellos, paquetes descartados por el Firewall, etc.
- *Configuración de los interfaces*. Muestra los principales valores de los interfaces, como por ejemplo su dirección IP, MAC, máscara, etc. También se muestran valores globales como la dirección del router por defecto, la del ACS (AutoConfiguration Server), el servidor SIP, servidores DNS, etc.
- *NAPT*. Aunque en el proyecto MUSE se deja claro que los usuarios residenciales no tendrán direcciones IP públicas en su hogar, lo que implica la utilización de NAPT en el RGW, nos parece interesante probar la ausencia del NAPT en el RGW, por lo que tenemos la opción de deshabilitar dicha funcionalidad.
- *Generador de tráfico*. Con esta funcionalidad queremos tener una herramienta que nos permita generar tráfico de una manera flexible. En este caso, podemos indicar la dirección IP origen, la IP destino, MAC origen y destino, el protocolo IP, el tamaño de la trama en bytes, el número de paquetes generados, el identificador de VLAN y si es un paquete de upstream o de downstream para probar diferentes puntos de inserción de tráfico.

#### 6.4. Pruebas realizadas

Se ha ideado un gran escenario de pruebas donde se probarán a la vez transmisiones de vídeo, voz sobre IP y tráfico web. Este se realizará con otros participantes del proyecto MUSE y hoy en día se siguen definiendo las diferentes pruebas a realizar.

Internamente se han realizado pequeñas pruebas para comprobar el correcto funcionamiento del RGW. Se ha utilizado un ordenador que simula los nodos de red (debe ser capaz de entender tramas VLAN) y dos ordenadores del usuario. En las pruebas el RGW obtiene su dirección IP WAN por una de sus interfaces y configura el resto de sus interfaces para finalmente ejecutar el servidor de DHCP e instalar el fichero Click correspondiente. Con esto se comprueba que el RGW se inicia sin problemas. Seguidamente se añaden flujos desde dentro o fuera de la casa para permitir salir los diferentes flujos de las pruebas. Se comprueba que los ordenadores del usuario tienen conectividad con la red de acceso realizando sesiones SSH. Se comprobó que no se pierden paquetes ni se pierde la conexión aunque se realicen altas o bajas de nuevos flujos.

#### 7. Conclusiones

En este artículo se ha presentado la arquitectura final de un RGW, resultado de un largo estudio llevado a cabo por varios expertos en la materia dentro del proyecto europeo MUSE. El punto principal ha sido la presentación del diseño y primeras pruebas de un prototipo de RGW basado en dicha arquitectura que cumple los requisitos de una solución intermedia entre un modelo Click puro y un modelo donde se desarrollasen todos los bloques funcionales a nivel de aplicación. De ahí surge el modelo híbrido que es una aportación novedosa de este trabajo y que ha sido validada por varias pruebas realizadas.

Otra de las aportaciones de este trabajo ha sido la demostración de que se puede desarrollar un RGW de grandes prestaciones utilizando un software especializado como en este caso es Click. Esta demostración se basa en la implementación de los distintos bloques funcionales utilizando tanto elementos ya disponibles en Click, como otros nuevos creados por nosotros. Como en cualquier prototipo, el primer paso fue el de buscar los elementos hardware y software óptimos para su desarrollo. La decisión de utilizar el software Click para el tratamiento de tramas a bajo nivel ha resultado ser muy adecuada así como también la idea de administrar el nivel Click mediante un interfaz web para facilitar su modificación.

Las pruebas finales demuestran que los objetivos marcados al principio del proyecto han sido cumplidos con creces y a tiempo, por lo que con esto se ratifica la viabilidad del prototipo del RGW dentro del proyecto MUSE.

#### Agradecimientos

Este trabajo ha sido parcialmente financiado por la Comisión Europea a través del proyecto MUSE.

#### Referencias

- [1] Multi Service Access Everywhere (MUSE) European Project, <http://www.ist-muse.org/>
- [2] DSL Forum. [Online] Available: <http://www.dslforum.org/>
- [3] Full Service-VDSL Committee. [Online] Available: <http://www.fs-vdsl.net/>
- [4] Click modular router. [Online] Available: <http://pdos.csail.mit.edu/click/>
- [5] IEEE 802.1Q, "Virtual Bridged Local Area Networks", 2003.