Clasificación de Flujos de Comunicación en Redes de 10 Gbps con FPGAs

Marco Forconesi[1,2], Gustavo Sutter[1], Sergio López-Buedo[1], Cristian Sisterna[2]
[1]Escuela Politécnica Superior, Universidad Autónoma de Madrid
[2]Facultad de Ingeniería, Universidad Nacional de San Juan
{marco.forconesi, gustavo.sutter, sergio.lopez-buedo}@uam.es; cristian@unsj.edu.ar

Resumen-- Este trabajo consiste en una aplicación hardware para medición y exportación de información relevante sobre el tráfico de datos en redes de ordenadores. En la industria es denominada NetFlow y la misma puede verse integrada en Routers comerciales de alta gama en el mercado. Sus aplicaciones son variadas y entre ellas se encuentran: monitoreo de redes, monitoreo de aplicaciones sobre redes, planificación de redes, análisis de seguridad, facturación, minería de datos, entre otras. Estas soluciones comerciales operan en enlaces de hasta 1 Gbps, pero a 10 Gbps solo lo implementan mediante muestreos de paquetes. El prototipo presentado reconstruye los flujos IP activos analizando todos los paquetes de la red.

Palabras clave-- NetFlow, NetFPGA, 10 Gbps Ethernet, FPGA, Packet Inspection.

I. INTRODUCCIÓN

En el contexto de las redes de ordenadores o redes IP, el análisis de tráfico con fines de administración y control se realiza, entre otros, construyendo un registro de los flujos IP activos en ciertos nodos de la red. Posteriormente esta información se exporta para su post-procesamiento. Con este objeto, Cisco Systems [1] desarrolló NetFlow [2] para la medición de tráfico mediante el análisis de los paquetes que circulan a través de los dispositivos que lo implementen, tales como Routers y Switches. La reconstrucción de los flujos IP activos se lleva a cabo internamente en estos dispositivos; luego esta información es exportada a través de una red a un colector remoto.

Los algoritmos de la capa de transporte más comúnmente utilizados para la exportación son: NetFlow v5, v9 [2] y el más reciente IPFIX [3]. IPFIX (IP Flow Information Export) es la estandarización de NetFlow v9. NetFlow v5 es el protocolo más difundido en la industria y es el que se escogió para implementar NetFlow a 10 Gbps en este trabajo.

El objetivo del presente trabajo es lograr el análisis del tráfico y construir los flujos sobre redes de 10 Gbps, para ello se utiliza una tarjeta basada en FPGAs de Xilinx (Virtex-5) con interfaces ópticas de red.

El presente artículo se organiza de la siguiente manera: la sección II describe la funcionalidad de NetFlow y la plataforma de desarrollo utilizada así como también el entorno de desarrollo. En la sección III una presentación de la arquitectura propuesta para la aplicación NetFlow a 10 Gbps. La sección IV expone la validación de resultados obtenidos y por último conclusiones y futuros trabajos en la sección V.

II. NETFLOW Y PLATAFORMA DE DESARROLLO

En esta sección se exponen los conceptos fundamentales sobre los que se basa este trabajo: NetFlow, NetFPGA-10G [4] y el entorno de desarrollo EDK [5].

A. NetFlow

Un flujo, según la definición de Cisco [2], es un conjunto de paquetes de información que son transmitidos de forma unidireccional entre un transmisor y un receptor. Cada flujo es identificado mediante una quíntupla (5-tupla de ahora en adelante) que se compone de:

- Dirección IP de origen
- Dirección IP de destino
- Puerto TCP/UDP de origen
- Puerto TCP/UDP de destino
- Protocolo de la capa de aplicación

La aplicación desarrollada analiza los paquetes que son recibidos por la interfaz de 10 Gbps y extrae los campos para formar la 5-tupla. En caso de tratarse de protocolos para fines logísticos de la red (como ARP, DHCP) el paquete es descartado. La 5-tupla recibida es buscada en un registro de flujos IP activos (memoria de flujos) para actualizar la información correspondiente a dicho flujo en caso de que éste estuviese presente. Si no se encontrase el flujo activo previamente, la 5-tupla recibida iniciará una nueva entrada en el registro de flujos activos. En la Figura 1 se muestra un diagrama correspondiente a la creación y actualización de los flujos activos en la memoria de flujos.

Además de la 5-tupla, de cada paquete se extrae la cantidad de bytes del campo IP Total Length el cual se va acumulando en la memoria de flujos por cada flujo, con el fin de determinar la cuantía de la información que el flujo transportó. En cada entrada de la memoria se almacena también, un contador de paquetes que contribuyeron a cada flujo, además se registra el tiempo de inicio y finalización del flujo (timestamp del primer y último paquete recibido). Por último, si el protocolo de la capa de aplicación es TCP, se guarda la OR-lógica de los flags de TCP. La Figura 2 muestra el contenido de una entrada en la memoria de flujos correspondiente a un flujo activo.

En paralelo a la creación y actualización de flujos, otro mecanismo de la aplicación lleva a cabo la exportación de los flujos cuando se cumpla alguna de las condiciones para que estos abandonen la memoria de flujos. Estas condiciones de expiración, son las siguientes:

- Superación del tiempo máximo de inactividad (*inactive timeout*). Típicamente de 15 segundos
- Superación del tiempo máximo de permanencia en la memoria de flujos (*active timeout*). Configurado en 30 minutos
- Conexión TCP finalizada o reseteada por flags de FIN o RST respectivamente.

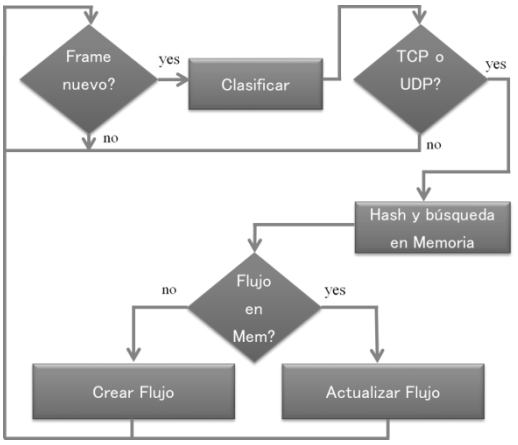

Figura 1: Creación y actualización de flujos

B. Plataforma de desarrollo NetFPGA-10G

NetFPGA es un proyecto de código software y hardware libre iniciado por Stanford University y Xilinx [6]. Consiste en una plataforma de hardware reconfigurable, FPGA Virtex-5 [7], dotada de interfaces Ethernet de 10 Gbps, con todos los elementos que éstas requieren para su funcionamiento; una interfaz PCIe para su conexión al computador, memorias RAM, puerto de comunicaciones RS-232, etc. El proyecto ofrece diseños funcionales de referencia para la puesta en marcha de la plataforma.

El objetivo del proyecto NetFPGA es crear las herramientas para el desarrollo rápido de dispositivos de redes basado en un apalancamiento en desarrollos previos. Esto último es posible gracias a que se sigue una metodología de diseño modular con interfaces estándares.

NetFPGA, al momento de realización del presente trabajo, tiene dos plataformas de desarrollo: NetFPGA-1G, la primera generación del proyecto y NetFPGA-10G, segunda generación, para la que existen desarrollos elementales y sobre la cual se trabajó para el desarrollo de esta aplicación hardware.

C. Entorno de Desarrollo

El proyecto NetFPGA-10G está basado en el flujo de diseño EDK [5] de Xilinx. Esto implica que existen Pcores con una determinada funcionalidad, interconectados a un procesador MicroBlaze [8].

El proyecto NetFPGA-10G utiliza el protocolo de interconexión AMBA® AXI4 [9] en todo su diseño. Los Pcores desarrollados para esta aplicación poseen estas interfaces para su integración en el proyecto.

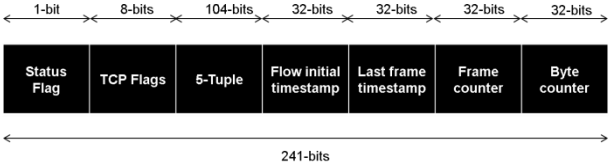

Figura 2: Contenido de la memoria de flujos por cada flujo activo

III. ARQUITECTURA PROPUESTA

Las restricciones temporales de este trabajo vienen dadas por completar todo el algoritmo de NetFlow a una velocidad igual o mayor a la velocidad de recepción de Frames de Ethernet a 10 Gbps. En el peor de los casos, con los Frames más cortos y el menor interframe gap contemplado en el estándar de Ethernet [10], se tienen 616-bits en total a una tasa de 10 Gbps, es decir 61,6 ns. La interfaz de usuario de los Pcores de acceso al medio 10GMAC [11], tienen una frecuencia de 200 MHz, lo que implica que cada 61,6 ns transcurren doce períodos de reloj. Como todo el hardware de NetFlow funciona con este mismo reloj, se cuenta con doce ticks para completar la operación y quedar disponible para el próximo Frame.

La aplicación NetFlow desarrollada está implementada con dos Pcores: NetFlow Cache y NetFlow Export. Siguiendo la arquitectura propuesta por Cisco [2], el primero construye y actualiza los flujos en la memoria de flujos y el segundo exporta, mediante NetFlow v5, los flujos expirados a un colector remoto. La Figura 3 muestra los Pcores de la aplicación NetFlow y su conexión a las interfaces de 10 Gbps mediante los Pcores 10GMAC. A continuación se describirá la funcionalidad de cada uno de los Pcores desarrollados así como también sus interfaces.

A. NetFlow Cache

Es el primer elemento en el camino de datos. Posee una interfaz AXI4-Stream esclava para la recepción de los Frames de Ethernet y una interfaz AXI4-Stream maestra para la exportación de los flujos expirados. Internamente posee la memoria de flujos, implementada con Block Ram de la FPGA, y dos procesos paralelos que actúan sobre esta memoria. El primer proceso es encargado de analizar los paquetes de la red para la construcción y actualización de los distintos flujos activos, utilizando para esto uno de los dos puertos de la memoria de flujos. El segundo proceso utiliza el otro puerto de la Block Ram y analiza continuamente si se ha cumplido alguna de las condiciones de expiración de algunos de los flujos. Cuando esto ocurre, el proceso quita el flujo de la memoria y exporta todo su contenido a una FIFO de salida. Mediante la interfaz maestra, los flujos expirados son leídos de esta FIFO y transmitidos al Pcore NetFlow Export.

El número de flujos que pueden ser albergados en la memoria de flujos es de 2^{14}. Una función de hashing sobre la 5-tupla reduce los 104-bits de ésta a 14-bits para direccionar la memoria de flujos y almacenar en dicha entrada el flujo correspondiente a la 5-tupla recibida.

El timestamping de los paquetes recibidos se lleva a cabo con un contador de milisegundos que comienza en cero al momento de la configuración de la FPGA. Una próxima versión incluirá un reloj sincronizado globalmente o incluso por señal GPS (Global Positioning System).
El proceso que analiza los paquetes, soporta VLAN y VLAN anidado, en caso de que estuviesen definidas redes virtuales. Esta capacidad permite que la aplicación NetFlow pueda trabajar en enlaces de redes donde se utilicen estos mecanismos.

B. NetFlow Export

Este Pcore implementa el protocolo de exportación NetFlow v5 con el fin de enviar a un colector remoto los registros de los flujos que se capturaron una vez que estos han expirado. Posee una interfaz esclava AXI4-Stream por donde recibe los flujos expirados enviados por NetFlow Cache. Otro par de interfaces AXI4-Stream, una esclava y otra maestra, se conectan a una interfaz Ethernet de 10 Gbps, a través de un Pcore de acceso al medio 10GMAC. NetFlow Export espera la llegada de treinta flujos expirados y envía un paquete con protocolo NetFlow v5 sobre UDP a la dirección IP del colector. Si ha recibido al menos un flujo expirado y la espera por un flujo para completar los treinta ha superado los sesenta segundos, NetFlow Export envía el paquete con el número de flujos que posea hasta ese momento.
La dirección IP del colector y de la aplicación NetFlow así como también la dirección MAC de esta última, son configuradas desde el EDK y no son accesibles por el MicroBlaze para su modificación. Si la dirección MAC del colector no es configurada desde el EDK, NetFlow Export, durante la inicialización, utiliza el protocolo ARP para determinar cuál es dicha dirección correspondiente a la IP del colector.

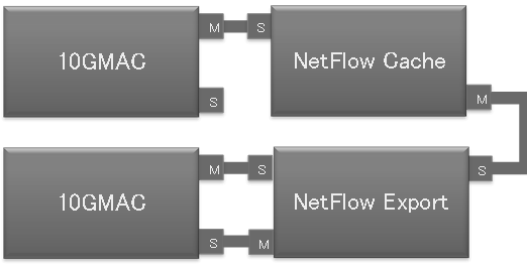

Figura 3: Pcores de aplicación NetFlow. Conexiónes AMBA AXI4-Stream

IV. RESULTADOS

El diseño fue realizado utilizando el lenguaje de descripción Hardware VHDL, sintetizados desde EDK 12.3 con Xilinx ISE/XST de la misma versión.
El funcionamiento de la aplicación NetFlow a 10 Gbps sobre la NetFPGA-10G ha sido validada utilizando réplicas de trazas sintéticas y captura de los paquetes NetFlow v5 que son enviados hacia el colector.
A continuación se describe el banco de pruebas utilizado y las verificaciones aisladas de cada uno de los Pcores. Por último la Tabla 1 muestra los detalles de uso de los recursos de la FPGA para cada uno de los Pcores y el sistema completo.

A. Arreglo experimental

Con el fin de inyectar tráfico Ethernet a 10 Gbps sobre la interfaz de captura de datos a la cual se conecta la aplicación NetFlow, se utiliza un computador con una NIC de 10 Gbps que se conecta mediante fibra óptica a un módulo SPF+ de una interfaz de la NetFPGA-10G.
En las trazas Ethernet que se inyectan, se incluyen muchos Frames con protocolos que no corresponden para NetFlow, como por ejemplo: ARP, ICMP, etc. Lógicamente, para corroborar que la aplicación NetFlow está funcionando correctamente, además de filtrar todos los Frames que no corresponden, debe capturar todos los flujos activos inyectados. Es necesario entonces, conocer cuáles son los flujos inyectados, para luego verificar que han sido procesados. Esto último se logra con trazas sintéticas de tráfico. El odenador que genera tráfico ejecuta el comando *tcpreplay* del bash de Linux para comenzar con la transmisión utilizando ficheros pcap con las trazas sintéticas.

B. Verificación de NetFlow Cache

En una primera etapa de validación, se verifica que la salida del Pcore NetFlow Cache cumple con los requisitos de diseño. Para esto, se conecta el Pcore al MicroBlaze mediante el protocolo de comunicación AXI4-Lite. El procesador ejecuta un programa de lectura de la FIFO de salida y muestra los flujos expirados en una consola de Linux mediante el puerto de comunicaciones RS-232. Se observa el tiempo transcurrido entre la inyección del tráfico y la salida de los flujos en la consola. Si es prácticamente instantánea la salida de un flujo, se debe a una finalización por los flags de TCP. Si transcurren 15 segundos es por superación del inactive timeout. Por último se ejecuta una inyección de paquetes de un mismo flujo por un tiempo prolongado, y se observa la salida del flujo de la memoria de flujos cada 30 minutos producido por la superación del active timeout.

Tabla 1: Resultados de implementación

	#Luts	#FF	#BRAMs
Sistema completo	24128 16%	30790 21%	142 43%
NetFlow Cache	1502 1%	1346 1%	122 37%
NetFlow Export	3954 2%	8509 5%	0

C. Verificación de NetFlow Export

Para la validación de NetFlow Export se conecta el sistema completo y se configura la FPGA. Una interfáz recibe el tráfico y otra envía los paquetes NetFlow v5 generados por la aplicación a partir de los flujos expirados. En el ordenador generador de tráfico, además del proceso que ejecuta *tcpreplay*, otro proceso ejecuta *tcpdump* para capturar los paquetes NetFlow v5 generados por la aplicación, en archivos que posteriormente son analizados con Wireshark [12].
Se han detectado flujos que no son capturados por NetFlow debido a las colisiones de la función hash por el reducido tamaño de la memoria de flujos. Un contador de colisiones codificado en el hardware

contabiliza la cantidad de veces que dos o más 5-tuplas diferentes intentan ubicarse en la misma dirección de memoria. En este caso el nuevo flujo es descartado conservándose el primero que ocupó dicha entrada.

V. Conclusiones y Futuros Trabajos

El prototipo de NetFlow, en su fase actual de desarrollo, es un módulo completamente operativo limitado a 2^{14} flujos concurrentes. El desarrollo permite clasificar tráfico de red a 10Gbps sobre la plataforma NetFPGA-10G (Virtex-5 TX240T).

Existen una serie de mejoras previstas con el fin de incrementar su rendimiento. Los troncales de 10 Gbps pueden tener, en condiciones extremas, decenas de miles de flujos concurrentes. Esto exige que la aplicación utilice memorias externas de manera de disponer de más capacidad para almacenar un mayor número de flujos activos. Esto último supone un desafío para cumplir las restricciones temporales disponibles para realizar el cómputo. Otra mejora planificada para la aplicación, con el fin de reducir la probabilidad de colisiones entre flujos concurrentes, es mejorar la función de hashing disponible actualmente, diseñando la más adecuada posible para distribuir de manera balanceada el espacio de entrada en la memoria de flujos disponibles. Por último es deseable tener la posibilidad de exportar los flujos expirados a través del bus PCIe [13] para realizar coprocesamiento así como también mediante una interfaz Ethernet utilizando el formato estándar de exportación IPFIX.

VI. Referencias

[1] Cisco Inc. http://www.cisco.com/, 2012.

[2] Cisco Inc., "NetFlow Services Solutions Guide," Tech. Rep., 2007. [Online]. Available: http://-www.cisco.com/en/US/docs/ios/solutions_docs/-netflow/nfwhite.html

[3] IETF (The Internet Engineering Task Force), *IP Flow Information Export (Active WG). Ipfix Status Pages*, The Internet Engineering Task Force (IETF) Std., 2012.

[4] NetFPGA Team, *NetFPGA-10G board description*, http://netfpga.org/, 2011.

[5] Xilinx Inc., *Embedded System Tools Reference Manual EDK 12.3. UG111*, 12th ed., http://www.xilinx.com/support/documentation, September 2010.

[6] Xilinx Inc. http://www.xilinx.com/, 2012.

[7] Xilinx Inc, *Virtex-5 FPGA Data Sheets*, http://www.xilinx.com/support/, March 2010.

[8] Xilinx Inc., *MicroBlaze Processor Reference Guide v11.2. UG081*, http://www.xilinx.com/support/, September 2010.

[9] ARM inc, "Amba axi protocol v2.0," Tech. Rep., 2010. [Online]. Available: http://www.arm.com/-products/system-ip/amba/amba-open-specifications.php

[10] IEEE Standard Association, *IEEE Standard for Information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements*, IEEE Std. 802.3bd-2011 (Amendment to 802.3-2008), 2008.

[11] Xilinx Inc., *LogiCORE IP 10-Gigabit Ethernet MAC v11.1. User Guide. UG773*, http://www.xilinx.com/support/, March 2011.

[12] G. Combs, *Wireshark Network Protocol Analyzer*, http://www.wireshark.org/about.html, 2012.

[13] PCI-SIG, *PCI ExpressTM Base Specification Revision 2.0*, http://www.pcisig.com/specifications/pciexpress/, January 2007.