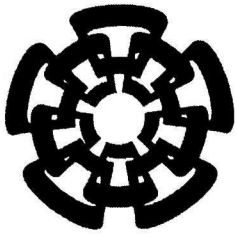
BC-660
Don.- 2011

xx (177964.1)



Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional
Unidad Guadalajara

Marco de Trabajo de Calidad de Servicio bajo un enfoque de comunicación fin a fin entre PCIe y Ethernet

Tesis que presenta:
Sagrario Corina Quevedo Pillado

para obtener el grado de:
Maestro en Ciencias

en la especialidad de:
Ingeniería Eléctrica

Director de Tesis
Dr. Mario Angel Siller González Pico

CINVESTAV del IPN Unidad Guadalajara, Guadalajara, Jalisco, Mayo de 2011

CLASIF.. 551-660
ADQUI
FECH 20 Octubre 2011
PROCED.. Don. - 2011
$

ID: 177395-1001

Marco de Trabajo de Calidad de Servicio bajo un enfoque de comunicación fin a fin entre PCIe y Ethernet

Tesis de Maestría en Ciencias
Ingeniería Eléctrica

Por:
Sagrario Corina Quevedo Pillado
Ingeniero en Sistemas Computacionales
Instituto Tecnologico de Tijuana 2001-2006

Becario de Conacyt, expediente no. 219314

Director de Tesis
Dr. Mario Angel Siller González Pico

CINVESTAV del IPN Unidad Guadalajara, Mayo de 2011

Resumen

Los protocolos de Calidad de servicio (QoS) tales como Intserv, Diffserv y MPLS son fundamentales en las transmisiones de tiempo real y aprovisionamiento de otros servicios en las redes de comunicaciones actuales. Esto se debe entre otras cosas a una mejor eficiencia en la utilización del ancho de banda, disminución de la perdida de paquetes y latencias que se pueden alcanzar.

Los protocolos de QoS se han visto limitados ya que normalmente se definen bajo una perspectiva de una red convencional y solo se ataca el problema tomando en cuenta los paquetes que se utilizan en dicha red, las prioridades y necesidades que tiene cada clase de tráfico. En lo investigado no se ha encontrado un trabajo de investigación en el cual analice redes de sistemas embebidos (intrasistemas) con redes convencionales (intersistemas) con el objetivo de relacionar sus características en común y poder brindar QoS bajo un enfoque fin a fin.

Esta investigación se enfoca en el diseño de un marco de trabajo de comunicación fin a fin que brinda QoS desde un PCI Express PCIe(intrasistema) hasta Ethernet(intersistema -LAN). El diseño se enfoca en extender las capacidades de QoS de los conmutadores PCIe hasta los conmutadores Ethernet.

Se realizaron simulaciones del marco de trabajo propuesto las cuales ayudaron a observar una mejora en el comportamiento de las métricas de QoS (latencia) con lo que se comprueba que se pueden extender las capacidades de QoS de PCIe a Ethernet.

Abstract

Nowadays, in real time transmissions quality of service has become a must due to it's different protocols like Intserv, Diffserv, MPLS that help to gain a better efficiency in bandwith usage, it lowers the lost rate of packets and latency. Providing QoS in computing and communications is currently the focus of many research efforts by industry and academy.

QoS has been limited because they are normally viewed from the perspective of a conventional network, and only attacks the problem by taking into account the packets that are used in the network and the priorities and needs of each traffic class. The investigation has not found a research job in which to analyze networks of embedded systems with conventional networks in order to relate their common characteristics and to provide QoS in a general way.

This research focuses on the design of a communication framework that provides end to end QoS from an intra (PCI Express PCIe) to an intersystem (Ethernet). The design focuses on extending the capabilities of PCIe switches QoS to Ethernet switches.

Simulations were performed proposed framework which helped to see an improvement in the behavior of QoS metrics (latency) with which it is found that can extend the capabilities of PCIe to Ethernet QoS.

Agradecimientos

A Dios por brindarme esta oportunidad de superación.

A mis padres por el apoyo incondicional en toda mi vida.

A la generación de computación 2008 por su gran compañia.

A mi asesor Mario Siller por su gran apoyo y enseñanza durante este tiempo.

A mis amigos que me ayudaban desde lejos.

A Conacyt por el apoyo económico.

Índice general

Capítulo 1

Introducción

Actualmente las exigencias en las transmisiones de datos en las redes computacionales han incrementado y demandan una diferenciación en las clases de tráfico para cumplir con determinados requerimientos. Calidad de servicio (QoS) brinda esta diferenciación de tráfico y así mismo contribuye a mejorar la calidad de experiencia del usuario mediente la aplicación de diferentes protocolos (Diffserv, IntServ, MPLS).

En este documento las redes computacionales cableadas de área local (Ethernet) son definidas como redes convencionales (intersistemas). En la mayoría de los casos se brinda QoS en las redes convencionales, pero, ¿solo se brinda QoS en intersistemas? la respuesta a la pregunta anterior es no, existen redes de sistemas embebidos (intrasistemas) que ya brindan QoS. Si ya se brinda QoS en redes convencionales y por otra parte en redes de sistemas embebidos, ¿Existe un marco de trabajo de comunicación fin a fin que nos ayude relacionar dichas redes y poder brindar QoS en forma general, a fin de obtener mayor eficiencia en la utilización del ancho de banda, menos tramas perdidas y menos latencia?

1.1. Descripción del problema

Bajo una perspectiva de trasmisión fin a fin entre nodos de una red de comunicación local (dominio local) extendida a la arquitectura de área local (Local Area Network LAN), metropolitana (Metropolitan Area Netowork MAN), amplia (Wide Area Network WAN) e Internet y protocolos de comunicación como PCI Express (PCIe), dentro del nodo terminal (dominio embebido), ¿es posible establecer modelos que permitan caracterizar el trafico de tramas?

Bajo este contexto, ¿existe alguna relación entre los mecanismos de calidad de servicio de ambos dominios (control de flujo, canales virtuales, clases de tráfico, protocolos) en cuanto a los efectos de su implementación a los principales parámetros de desempeño de comunicación fin a fin, tales como eficiencia de transmisión, latencia y perdida de tramas?

Las redes heterogéneas, la mayoría de las veces, trabajan independientemente, desaprovechando la posibilidad de trabajar y aplicar diferentes mecanismos de manera global, sin tener la necesidad de emplearlos en cada dominio de la red. En PCIe los paquetes son marcados por el conmutador, dependiendo el canal virtual al que pertenece, esta marca se pierde al entrar a la red Ethernet, donde, se vuelve a marcar para volver a viajar por la red. Debido a la necesidad de nuevos marcados en cada transmisión, no resulta adecuada esta manera de configuración de red, por lo que ¿será necesario definir un marco de trabajo, para lograr la interoperabilidad entre la tecnología PCIe y Ethernet, disminuyendo el número de marcado de paquetes?

1.2. Objetivos

1.2.1. Generales

El objetivo de esta investigación es estudiar la interoperabilidad de los mecanismos de QoS de PCIe y Ethernet en una perspectiva fin a fin para mejorar la eficiencia de transmision y reducir latencias y tramas perdidas (Fig. 1.1).

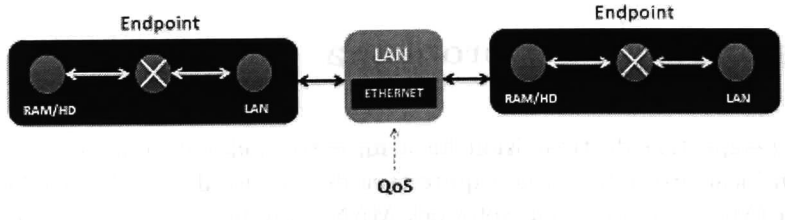


Figura 1.1: Marco de trabajo de comunicación fin a fin

1.2.2. Particulares

- Analizar detalladamente las características de la red de PCIe y la red Ethernet, con el fin de encontrar similitudes en dichas características bajo el contexto de QoS (Clases de tráfico, Prioridades, Control de flujo).
- Analizar detalladamente la interaccion entre los protocolos de PCIe y Ethernet con el fin de conocer su funcionamiento y determinar los campos de mejora.
- Desarrollar un marco de trabajo de comunicación fin a fin resultante del análisis comparativo entre la red PCIe y Ethernet, al mismo tiempo crear entidades que sean necesarias para poder cumplir con la función del marco de trabajo.
- Simular el marco de trabajo creado en NS3 (network simulator) con el objetivo de validarlo y capturar datos (tiempo de transmisión, retardos, perdida de tramas) para hacer una comparación con otros trabajos y verificar que en efecto se mejoraron las métricas de QoS tales como el rendimiento, las tramas perdidas y la latencia (Figura 1.2).

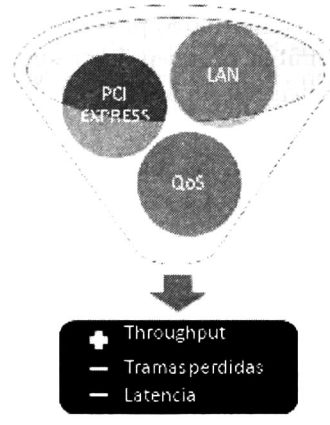


Figura 1.2: Métricas de QoS

1.3. Estructura de la tesis

La tesis es organizada en 5 capítulos: marco teórico, propuesta, simulación de la propuesta, resultados y análisis y conclusiones y trabajo futuro.

La sección de marco teórico (capítulo 2) presenta una revisión de los conceptos teóricos relacionados con la investigación. Esta revisión abarca conceptos básicos de redes, intersistemas e intrasistemas finalizando con conceptos precisos de tecnologías como PCI Express (PCIe) y Ethernet bajo el contexto de Calidad de Servicio (QoS).

La sección de la propuesta (capítulo 3) propone un marco de trabajo que ofrece Calidad de Servicio (QoS) en una comunicación fin a fin desde PCIe hasta una red Ethernet. Éste consta de tres partes fundamentales: control de flujo, manejo de congestión y servicios diferenciados. En cada una de éstas se describen los mecanismos que cada tecnología (PCI Express y Ethernet) van a utilizar, de igual manera se describen las relaciones que ocurren entre las clases de tráfico de cada una de ellas.

La sección de simulación del marco de trabajo (capítulo 4) muestra una descripción, diseño y código utilizado para desarrollar un simulador del marco de trabajo propuesto. Este simulador se utilizó para realizar pruebas generales a efecto de demostrar que los mecanismos propuestos mejoran el rendimiento de la red.

La sección de análisis y resultados (capítulo 5) presenta una serie de simulaciones y análisis de casos de estudios, que nos permiten observar los resultados obtenidos al implementar nuestro marco de trabajo de comunicación fin a fin desde PCI Express hasta Ethernet.

Por último en el capítulo 6 se muestran las conclusiones a las que llegamos y una lista de trabajos que surgen para complementar esta tesis.

Capítulo 2

Marco Teórico.

En este capítulo se presenta una revisión de los conceptos teóricos relacionados con la investigación. Esta revisión abarca conceptos básicos de redes, intersistemas e intrasistemas finalizando con conceptos precisos de tecnologías como PCI Express (PCIe) y Ethernet bajo el contexto de Calidad de Servicio (QoS). Es necesario conocer estos conceptos para comprender totalmente el contenido de la tesis.

2.1. Redes

Una de las formas de comunicación más usadas en la actualidad son las redes de comunicaciones. Una red es un conjunto de dispositivos conectados por enlaces de un medio físico [1].

Las redes se clasifican según su tamaño (Figura 2.1) en 6 diferentes tipos: Internet, red de área amplia (Wide Area Nerkwork WAN), red de área metropolitana (Metropolitan Area Network MAN), red de área local (Local Area Network LAN), red de área personal (Personal Area Network PAN), circuito impreso (Printed Circuit Board PCB) y redes en chip (Network on Chip NOC).

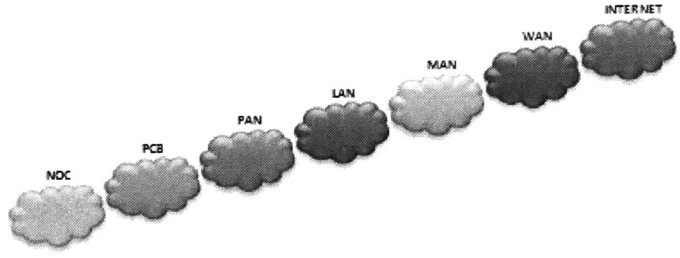


Figura 2.1: Clasificación de redes según su tamaño.

En nuestro trabajo le llamaremos intersistemas a todas los sistemas que se relacionen con otros sistemas(WAN, MAN, LAN y PAN) e intrasistemas a los sistemas que solo se relacionan con ellos mismos (PCB).

Nuestra investigación se centra en la extensión de las capacidades de QoS de un intrasistema (PCI Express) a un intersistema (Ethernet).

2.2. Intrasistemas (PCI Express)

Los PCB se utilizan para soporte y conexión de componentes electrónicos utilizando vías conductoras, pistas o huellas grabadas de las hojas de cobre laminadas sobre un sustrato no conductor. También se le conoce como placa de circuito impreso (PCB) o placa de circuito grabado [2].

Uno de los ejemplos más importantes de los PCB es PCI Express (PCIe) [3]. PCIe es la tercera generación tecnología para interconectar dispositivos periféricos e incorpora en sus últimos avances alta velocidad e interconexiones punto a punto. PCIe ofrece un rendimiento significativamente superior, confiabilidad y capacidades mejoradas a un costo menor que las versiones de PCI.

PCIe ofrece varios beneficios: alto rendimiento, simplificación en entrada y salida (E/S), una arquitectura en capas, siguiente generación multimedia, facilidad de uso y QoS.

El alto rendimiento de E/S es utilizado para interconectar dispositivos periféricos en aplicaciones tales como plataformas informáticas y de comunicación [3]. PCIe es serial por lo que la comunicación entre sus dispositivos es punto a punto.

Asi mismo PCIe implementa la tecnología basada en conmutación de paquetes para interconectar un gran número de dispositivos. La comunicación a través de la interconexión en serie se realiza mediante un protocolo de comunicación.

PCIe esta compuesto por varios tipos de dispositivos: (1) "Root complex", (2) "Endpoint" (3)"Switch" y (4) "PCI Express to PCI bridge (Figura 2.2). "Root complex" es la raíz de las conexiones del sistema de E/S al CPU y memoria. "Endpoint" en un dispositivo que puede solicitar/completar una transacción PCIe por el mismo o en nombre de un dispositivo que no sea PCIe. El "Switch" se encarga de actuar como el director de tráfico entre los múltiples enlaces. El "PCI Express to PCI bridge" tiene un puerto PCI Express para uno o múltiples interfaces bus PCI/PCI-X, y se encarga de que PCIe pueda coexistir con la tecnología existente de PCI.

Los dispositivos PCIe se comunican por medio de enlaces. Un Enlace (E) es una colección de líneas de comunicación entre dos dispositivos PCIe. Un enlace PCIe cuenta hasta con 32 líneas (x1, x2, x4, x8, x16 y x32):

$$E = \{L_1, \ldots, L_{32}\}$$

Una Línea (L) está formada por un conjunto de pares Transmisión (Tx) y Recepción (Rx) (Fig. 2.3). Cada L contiene 4 señales (S):

$$S \in L$$

$$L = \{S_1, S_2, S_3, S_4\}$$

$$L = \{Tx, Rx\}$$

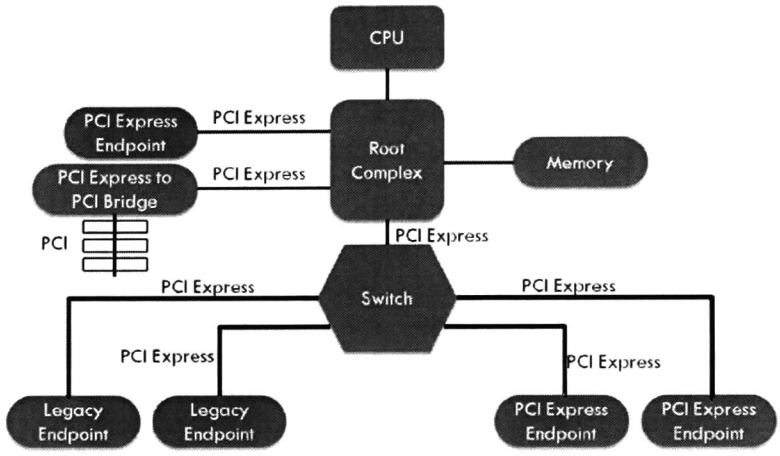


Figura 2.2: Arquitectura PCIe [6].

$$TX = \{S_1, S_2\}$$

$$RX = \{S_1, S_2\}$$

$$S1 = T+$$

$$S2 = T-$$

$$S3 = R+$$

$$S4 = R-$$

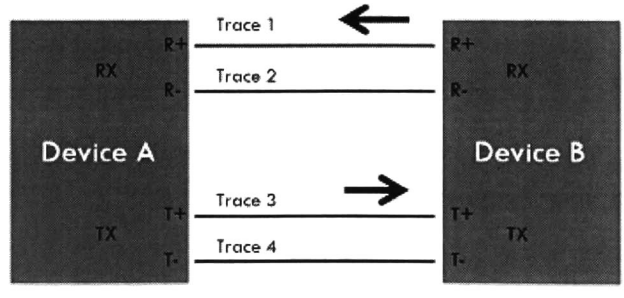


Figura 2.3: Conexión punto a punto entre dos dispositivos PCI Express de una Línea.

Existen varias versiones de PCIe (1.x, 2.x, y 3.0), siendo la 3.0 la versión reciente (2010). PCIe 3.0 duplica la tasa de datos de PCIe 2.x a 8 GT/s, mientras que PCIe 2.x ya había duplicado la tasa de datos de PCIe 1.x. Así mismo se duplica el ancho de banda de la interconexión de cada versión con respecto a su versión anterior. También existe un cambio en el sistema de codificación. PCIe 1.1 y PCIe 2.0 utilizan el sistema de codificación 8b/10b mientras que PCIe 3.0 incluye un nuevo sistema de codificación 128b/130b (Tabla 2.1) ofreciendo una aumento en su eficiencia.

Arquitectura PCIe	Velocidad de datos	Ancho de banda de la interconexión	Ancho de banda por línea por dirección	Ancho de banda total en un enlace x16
PCIe 1.x	2.5GT/s	2Gb/s	~250MB/s	~8GB/s
PCIe 2.x	5.0GT/s	4Gb/s	~500MB/s	~16GB/s
PCIe 3.0	8.0GT/s	8Gb/s	~1GB/s	~32GB/s

Tabla 2.1: Ancho de banda y velocidad de datos de las versiones de PCIe 1.x 2.x y 3.0.

2.2.1. Calidad de Servicio

Calidad de Servicio (Quality of Service QoS) es un término genérico que se refiere normalmente a la capacidad de una red u otra entidad (por ejemplo PCIe) para proporcionar latencia predecible, que es el número de pulsos que toma a un controlador de PCIe procesar un paquete. Así también ancho de banda predecible que es la tasa en la cual los datos pueden ser transmitidos o recibidos sobre un enlace [3].

QoS es de particular interés cuando las aplicaciones requieren ancho de banda garantizado en intervalos regulares, como los flujos de audio. Para ayudar a lidiar con este tipo de requisitos, PCIe define las transacciones isócronas, que requieren un alto grado de calidad de servicio. Sin embargo, QoS puede aplicarse a cualquier transacción o serie de operaciones que utilizan la estructura de PCIe.

QoS incluye los siguientes elementos para medir rendimiento de PCIe: (1) tasa de transmisión, (2) ancho de banda efectivo, (3) latencia y (4) tasa de error. PCIe necesita de los siguientes elementos para poder brindar QoS:

Clases de Tráfico (TC)

Canales virtuales (VC)

"Port Arbitration"

- "VC Arbitration"

  Control del flujo del enlace

Estos elementos se definirán posteriormente. PCIe utiliza dos clases generales de transacciones las cuales se benefician de la aplicación de QoS:

- Transacciones isocrónica: de iso (mismo) + síncrona (tiempo), éstas operaciones requieren un ancho de banda constante en intervalos regulares con una latencia garantizada.
- Transacciones asincrónicas: Esta clase de transacciones involucra una gran variedad de aplicaciones que tienen una amplia diversidad de requisitos de ancho de banda y latencia.

2.2.2. Clases de Tráfico y Canales Virtuales

El valor de la Clase de Tráfico (TC) se especifica en la cabecera del paquete de transacción y pueden contener uno de los ocho valores (TC_0-TC_7). TC_0 deben ser aplicadas por todos los dispositivos PCIe, los valores de TC_1-TC_7 son opcionales y proporcionan siete niveles de arbitraje para diferenciar entre flujos de paquetes que requieren diferentes cantidades de ancho de banda. Del mismo modo, se especifican ocho Canales Virtuales (VC) (VC_0-VC_7), VC_0 obligatorios y VC_1-VC_7 opcionales:

$$PCIe = \{VC_0, \ldots, VC_7\}$$

Existen transacciones $PCIe$ que solo pueden utilizar TC_0 relacionado con su VC_0, estas son:

- "Configuration"

  "I/O"

- "INTx Message"

"Power Management Message"

- "Error Signaling Message"
- "Unlock Message"
- "Set_Slot_Power_Limit Message"

Los dispositivos PCIe pueden soportan diferentes números de VC. Para determinar el valor del VC a utilizar, el software verifica el número de VCs soportados por los dispositivos conectados a un enlace común y le asigna el mayor número de VCs que los dispositivos tienen en común.

2.2.3. Arbitraje

Siempre que existe un recurso, como un "buffer", un canal o un puerto de un conmutador, es compartido por muchos agentes. Un árbitro está obligado a asignarle el recurso a un agente a la vez. Existen diferentes mecanismos para brindar arbitraje según sea necesario [4]:

"Arbitration timing": La duración del servicio depende de la aplicación ya que se le asigna un tiempo determinado de servicio.

"Fairness": Tienen la propiedad de ser justo. Intuitivamente, un árbitro justo provee el mismo servicio a los diferentes solicitantes del recurso.

"Fixed priority arbiter": Trabaja por medio de prioridades, atiende a los de más alta prioridad, de no haber solicitantes de esa prioridad continua con el siguiente en prioridad y así sucesivamente.

"Variable proprity iterative arbiters": Se pueden utilizar arbitrajes iterativos justos en los que se puede cambiar la prioridad del cliente de un ciclo a otro.

- "Oblivious arbiter": Este mecanismo opera brindando servicios aleatorios por lo que no son muy justos.
- "Round Robin": Opera sobre el principio en el cual la solicitud del cliente que se sirvió debe tener la menor prioridad que los demás en la próxima ronda de arbitraje.

- "Grand-hold circuit": Un cliente que recibe un concesión puede utilizar el recurso para un solo ciclo y debe arbitrar de nuevo para el uso del recurso.
- "Weighted round-robin": Algunas aplicaciones requieren un árbitro que sea un cierto grado de injusticia, de modo que un solicitante recibe un número mayor de concesiones que otro solicitante. A cada solicitante se le asigna un peso que indicará las concesiones que tendrá sobre el recurso.

- Queuing Arbiter: Provee una prioridad que consiste en que el primero que solicita el recurso es el primero a que se le asigna (First In First Out FIFO).

El uso de QoS se basa en una aplicación de software para la asignación TC que definen la prioridad de cada transacción. A cada TC se le asigna a un VC que se utiliza para gestionar la prioridad de transacciones a través de dos regímenes de arbitraje implementados por los conmutadores "Port arbitration" y "VC arbitration" (Figura 2.4).

"Port Arbitration": Es el arbitraje de dos paquetes que llegan de diferentes puertos de entrada pero se les asigna el mismo canal virtual. El "Port Arbitration" implementa "round robin" WRR (weight round robin) y "round robin" basado en tiempo.

"VC Arbitration": Toma lugar después del "Port Aritration" Los paquetes de todos los VC compiten para transmitir sobre el mismo puerto de salida. "VC Arbitration"Determina la prioridad de transacciones que se transmiten desde el mismo puerto en base a su VC [1].

El arbitraje en el switch de PCIe funciona de la siguiente manera:

1. Los paquetes arriban a los puertos de ingreso (1-256), y son colocados en su correspondiente buffer de control de flujo receptor del canal virtual al que corresponden.
2. Cada paquete contiene información de ruteo en donde incluye información del puerto de egreso objetivo.
3. Por cada paquete del puerto de ingreso se le asigna a su respectivo puerto de egreso dependiendo su canal virtual.
4. Un "Port Arbitration"se utiliza por cada ID de los canales virtuales y dependiendo del mecanismo ("fixed" o "WRR") los paquetes son enviados a otro buffer donde un "VC arbitration" utilizará a su vez un mecanismo para seleccionar el paquete que utilizará el puerto de egreso.

El mecanismo de QoS de servicios diferenciados trabaja marcando cada paquete para formar diferentes clases de tráfico y cada clase recibe diferente servicio.

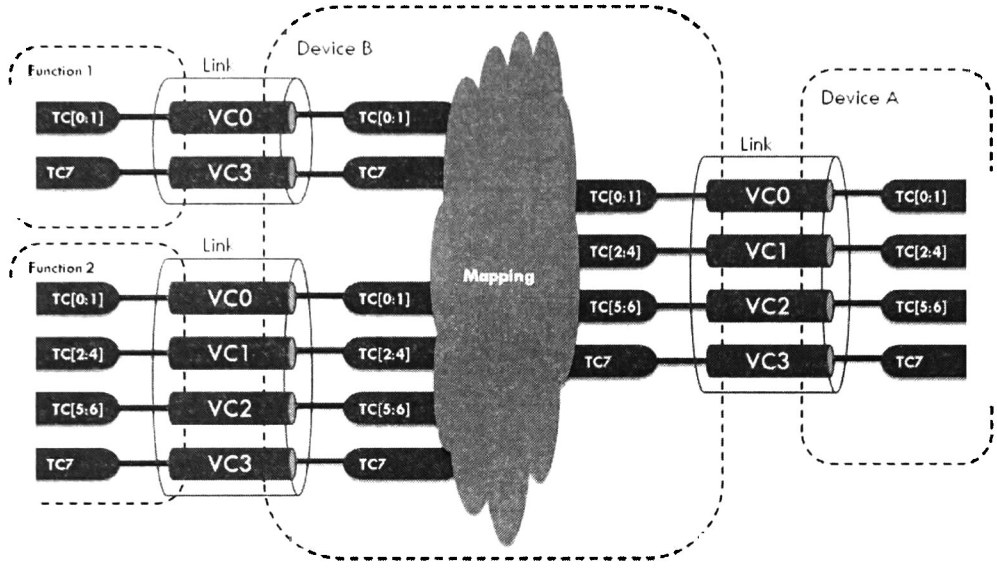


Figura 2.4: Clases de tráfico y canales virtuales en PCI Express.

2.2.4. Control de Flujo y Manejo de Congestión

Existen tres tipos de mecanismos de control de flujo: (1) Basado en créditos. (2) On/Off y (3) Ack/Nack.

En el control de flujo basado en créditos, el "router" "upstream" mantiene la cuenta del número libre de espacios en el "buffer" "downstream" de cada canal virtual. Cada vez que el "router" "upstream" envía un paquete, se consume un crédito del "downstream" y se decrementa de la cuenta. Si la cuenta llega a cero, esto indica que todos los "buffers" del "downstream" están llenos y no se pueden mandar paquetes hasta que estén disponibles de nueva cuenta los buffers. Una vez que el router "downstream" libera un espacio en el "buffer". envía un crédito al "router" del "upstream" incrementando la cuenta de éste (Figura 2.5) [4].

Por otra parte en el control de flujo On/Off el estado del "upstream" es un solo bit de control que representa cuando al nodo "upstream" se le permite enviar (on) o no (off). Una

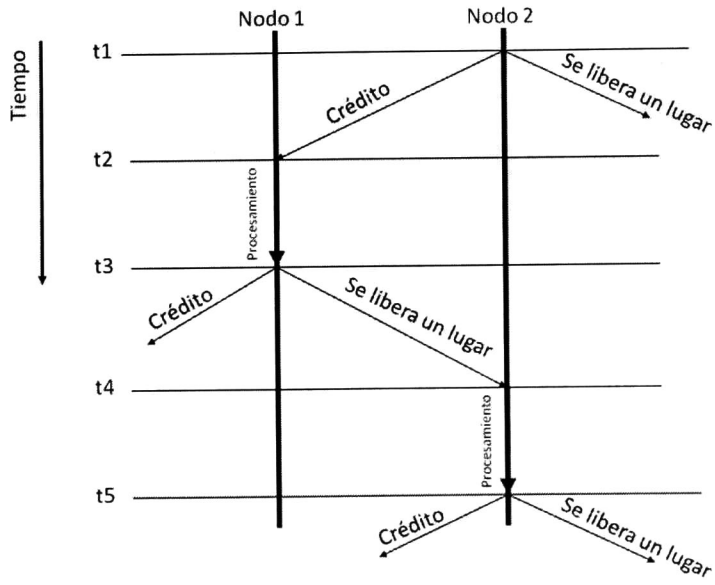


Figura 2.5: Línea de tiempo del mecanismo de control de flujo basado en créditos.

señal es enviada al "upstream" solo si es necesario cambiar de estado. La señal de "off" es enviada cuando el bit de control es "on" y el número de lugares libres del "buffer" son menores que la cota Foff. Si el bit de control es "off" y el número de espacios libres del buffer está por encima de la cota Fon, la señal de "on" es enviada (Figura 2.6).

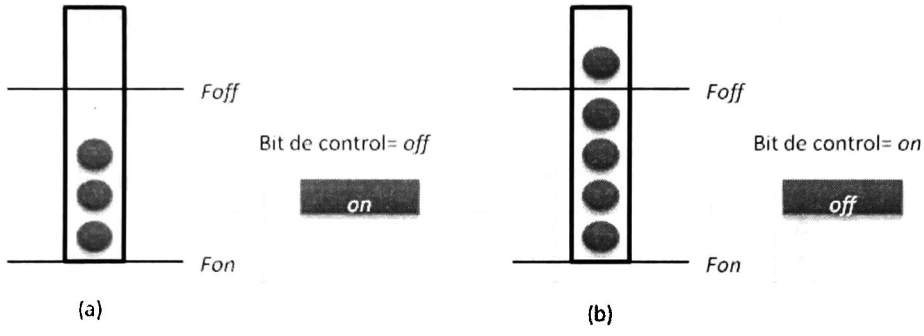


Figura 2.6: Mecanismo de control de flujo Xon/Xoff.

El control de flujo Ack/Nack trabaja mandando paquetes por parte del "upstream", si el "buffer" del nodo de "downstream" tiene espacios disponibles, acepta el paquete y manda un "acknowledge" (Ack) al nodo upstream. Si el buffer del nodo downstream no tiene espacios disponibles manda un "acknowledgment" negativo (Nack) al nodo "upstream" (Figura 2.7). Éste mecanismo es menos efectivo que el basado en créditos y el On/Off ya que manda paquetes sin cersiorarse que exista espacio suficiente en el "buffer" del "downstream" lo que hace que pierda demasiados paquetes, por lo tanto el uso del ancho de banda es ineficiente.

PCIe requiere que cada dispositivo implemente un control de flujo del enlace basado en crédito para cada canal virtual en cada puerto. Para las garantías de control de flujo los transmisores no enviarán paquetes de la capa de transacción (TLP) que el receptor no pueda aceptar. Por ésta razón se usa el control de flujo basado en créditos, primero se verifica si el nodo "downstream" tiene suficientes créditos para el paquete si es asi el nodo "upstream" procede a enviar el paquete, en caso contrario, espera a que el nodo "downstream" tenga créditos disponibles para poder mandar dicho paquete. Dado que PCIe es punto a punto y el control de flujo se da por cada enlace, no se necesita otro mecanismo para controlar la congestión ya que el mismo control de flujo basado en créditos ayuda a controlar dicha congestión debido a la limitante de no enviar paquetes si no hay créditos disponibles.

El manejo de la congestión en PCIe se da implícitamente gracias a que su comunicación es punto a punto, por tanto, si se terminaran los créditos ya no se pueden enviar paquetes, razón por la cual no tendrían congestión.

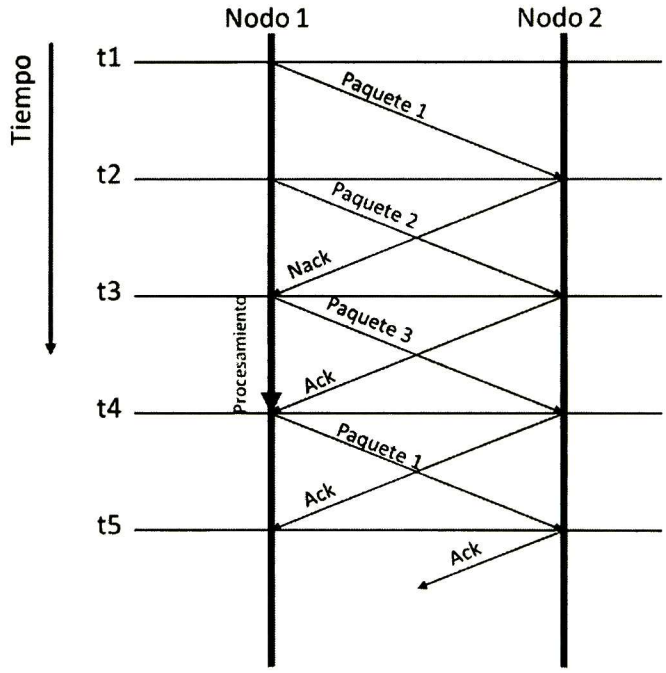


Figura 2.7: Línea de tiempo del mecanismo de control de flujo Ack/Nack.

2.3. Intersistemas (Ethernet)

Las redes LAN suelen ser privadas, ya que normalmente se usan en empresas y cada empresa determina su propia red. Las redes LAN pueden ir desde una red con dos computadoras hasta una empresa que utilice audio/video. Actualmente estas redes se limitan a unos pocos kilómetros.

2.3.1. Ethernet

Ethernet es un estándar de redes de computadoras de área local con acceso al medio por contención CSMA/CD ("Acceso Múltiple por Detección de Portadora con Detección de Colisiones"), es una técnica usada en redes Ethernet para mejorar sus prestaciones. Ethernet define las características de cableado y señalización de nivel físico y los formatos de tramas de datos del nivel de enlace de datos del modelo OSI.

Ethernet se tomó como base para la redacción del estándar internacional IEEE 802.3. Usualmente se toman Ethernet e IEEE 802.3 como sinónimos. Ambas se diferencian en uno de los campos de la trama de datos (Figura 2.8). Las tramas Ethernet e IEEE 802.3 pueden coexistir en la misma red [5].

2.3.2. Calidad de Servicio

Calidad de Servicio (QoS) se brinda en la capa 3 (capa de red) del modelo OSI. Sin embargo se puede adaptar la capa 2 para ayudar a brindar QoS. QoS cuenta con una variedad de mecanismos (servicios integrados, servicios diferenciados, etc) que nos ayudan a mejorar la entrega de datos en una comunicación fin a fin. Entre los mecanismos más importantes destacan [1]:

Servicios integrados (intserv)

Los recursos de la red se distribuyen de acuerdo a las aplicaciones que solicitan QoS y se basa a la política de gestión de ancho de banda.

Intserv describe tres tipos principales de servicio que puede solicitar una aplicación:

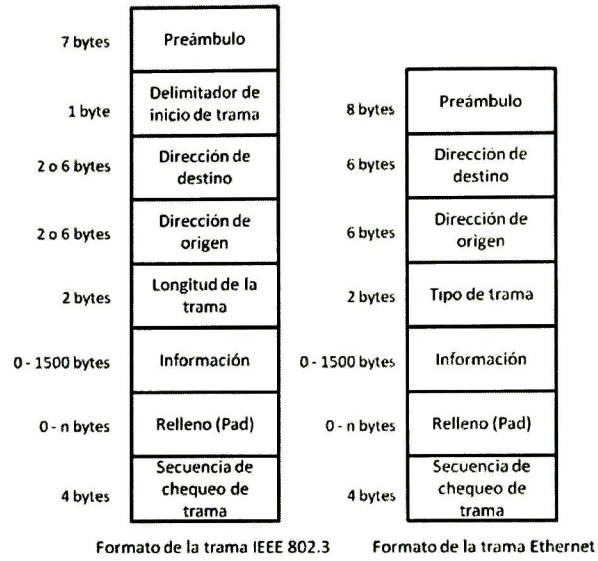


Figura 2.8: Formato de las tramas Ethernet e IEEE 802.3.

- Servicio garantizado (Guaranteed Services) [6]: Provee límites firmes (matemáticamente probables) sobre las demoras de encolado de paquetes fin a fin, haciendo lo posible por proveer un servicio que garantice el ancho de banda y la demora.

- Carga controlada (Controlled load) [7]: Provee al flujo de aplicación una calidad de servicio equivalente al QoS que recibiría el mismo flujo en un elemento de red que se encuentra bajo muy poca carga, pero utiliza control de capacidad (adminisión) para asegurar que el servicio es recibido aún si el elemento de red está sobrecargado.

- Servicio de mejor esfuerzo (Besteffort service): No provee ninguna garantía de servicio de ningún tipo.

- **Servicios diferenciados (diffserv)**

  Los servicios diferenciados constituyen un modelo de QoS basado en clases diseñado para IP. La cabecera IP contiene un campo de tipo de servicio (Type of Service ToS). Las aplicaciones pueden configurar tres bits de precedencia del campo ToS en el nivel de la tarjeta de interfaz de red (NIC) de acuerdo a sus necesidades.

Para Diffserv (RFC 2474) definen un campo especial para diferenciación de servicios (DS) en las cabeceras de IPv4 e IPv6. La estructura del campo se presenta en la Figura 2.9 [8].

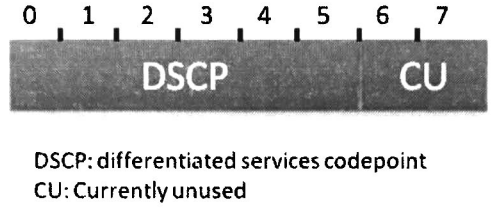


Figura 2.9: Campo especial para la diferenciación de servicios.

El tráfico de la red se clasifica y se distribuye entre los recursos de la red en función de las políticas de administración del ancho de banda. Para habilitar la QoS, los elementos de la red que sean identificados como más exigentes deben de recibir un trato preferencial.

Se pueden construir servicios significativos con la combinación de lo siguiente:

- Activando un campo de la cabecera IP al momento de entrar en la red o en los límites de la red (IPP o DSCP).
- Usando este campo para determinar en los nodos dentro de la red el reenvío de paquetes.
- Condicionando a los paquetes marcados en los límites de la red de acuerdo con los requerimientos o reglas de cada clase o servicio.

Estos mecanismos pueden usarse individualmente o combinándolos para formar diferentes arquitecturas con la finalidad de dar solución a un determinado problema (entrega de tramas en un determinado tiempo, aumentar el rendimiento, disminuir retardos, etc).

Se creará una arquitectura que cubra características en común entre las redes convencionales y las redes de sistemas embebidos y a dichas características se les brindara QoS. Al atacar el problema desde el inicio de la comunicación mejorará el desempeño de la red.

2.3.3. Control de Flujo y Manejo de Congestión

Ethernet utiliza un mecanismo de control de flujo basado en tiempo Xon/Xoff. Éste trabaja conjuntamente con el mecanismo de manejo de congestión en Ethernet (Ethernet

Congestion Management ECM).

El problema de la congestión toma lugar si el total de la suma de las demandas sobre un recurso es mayor que la capacidad de disponibilidad de éste. Representado matemáticamente:

$$\Sigma \text{Demanda} > \text{Recursos disponibles [9]}$$

Ante este problema en la tecnología Ethernet, existen diferentes mecanismos de control de congestión entre los que destaca el ECM [10]. ECM es un mecanismo capa 2 de manejo de congestión, anteriormente conocido como Notificación de Congestión Hacia Atras (Backward Congestion Notification BCN). Éste mecanismo consiste en llevar la congestión del núcleo hasta los extremos de las red, usar limitadores de tasas para los flujos causantes de la congestión y controlar la tasa de inyección basado en "feedbacks" provenientes de los puntos de congestión. ECM trabaja de la siguiente manera:

1. Detección: Los puntos de congestión son monitoreados, se verifica que las cotas de las colas no sean excedidas.

2. Señalización. El punto de congestión (donde la congestión es detectada) notifica al nodo final que existe una congestión. La notificación es definida por mensajes BCN.

3. Reacción. Se ajusta los limites de las tasas de los puntos de reacción de acuerdo a los mensajes BCN recibidos. Si el punto de reacción recibe un negativo "feedback" ($FB < 0$), éste disminuye la tasa de inyección de tráfico, si se recibe un positivo "feedback" se incrementa la tasa de inyección de tráfico.

2.4. Trabajo Relacionado

Actualmente se enlazan muchos tipos de redes para solucionar un problema determinado, a la combinación de estas redes se le conoce como redes heterogéneas. Las redes heterogéneas surgen por la necesidad que tienen las personas de comunicarse con diferentes tipos de dominios de red. Con el fin de interrelacionar y lograr una comunicación eficiente entre los diferentes dominios de redes se requiere definir una infraestructura junto con todas las características fundamentales que sirven de referencia para enfrentar y resolver problemas de índole similar, a esto se le conoce como marco de trabajo.

Existen diferentes tipos de marcos de trabajo enfocados a determinados casos de redes heterogéneas, entre los que destacan los que abordan el tema de QoS. Existen trabajos que

afirman que para brindar QoS en una comunicación fin a fin, es necesaria de una arquitectura que gestione los puntos finales [11].

Algunos trabajos se proponen funciones que ayudan a brindar QoS, entre las más importantes resalta la negociación. Esta negociación se hace entre dispositivos primarios y puntos finales y por los recursos compartidos (ancho de banda, memoria, dispositivo de entrada salida).

En la mayoría de los trabajos se proponen mecanismos de manejo de congestión y control de flujo con el objetivo de ofrecer QoS, algunos de los cuales se describen a continuación.

A efecto de ofrecer QoS en una comunicación fin a fin, se necesita tomar en cuenta los retardos en cada uno de los dispositivos (ruteadores) involurados. Hay ocasiones en donde un paquete tiene que pasar por varios ruteadores sufriendo retardos adicionales y cayendo en el problema de la "congestión transitoria" Existen trabajos tales como [12] que aplican un mecanismo de control de congestión llamado "Control de Flujo Basado en el Estado" (Status Based Flow Control SBFC) para resolver el problema de la congestión transitoria. SBFC trabaja en base a notificaciones de acuerdo al grado de congestión del router.

En el trabajo [13] extiende el soporte de QoS de la capa 3 a la capa 2. Este soporte se da por la cooperación de las dos capas. Según el estándar 802.1p que habla de proveer QoS en conmutadores Ethernet, define arquitecturas de puentes virtuales de redes LAn (VLANs), en la cual se usan tramas marcadas para poder realizar la diferenciación de clases.

En la capa dos utilizaron clasificación, marcado de L2 y políticas. Y en la capa 3 (Re) marcado, clasificación, políticas y organización. Se realizaron experimentos en los que se observó que con la cooperación de las dos capas para brindar QoS se vieron beneficiadas las clases de tráfico de mayor prioridad.

Hay otros trabajos en donde utilizan cruce de capas. El [14] optimiza el soporte de QoS en sistemas DVB-RCS (Digital Video Broadcasting - Return via Satellite). Anteriormente los sistemas DVB-RCS utilizaban 6 colas en la capa 3 y Diffserv. En la capa 2 los paquetes eran segmentados y encapsulados, utilizaban 3 colas y un campo DS para clasificar el tráfico (que es el campo de la capa 3). Como ésto involucraba algunos mecanismos duplicados, el [14] reduce el tamaño de las colas en la capa 2 sin ninguna pérdida, así mismo reducen el numero de colas de la capa 2 (una sola cola). Esto lo logran a través de mecanismos de cruce de capas. El primer mecanismo es el de tener conocimiento del ancho de banda (por la capa 3) disponible en la capa 2. El segundo es el de tener información de los mecanismos de encapsulamiento y segmentación de la capa 2, necesario para regular las salidas del planificador de la capa 3. Al aplicar estos mecanismos obtuvieron una disminución en la latencia.

En la literatura revisada se encontraron muchos trabajos donde utilizan la capa 2 y capa 3 en cooperación para brindar QoS, sin embargo en la revisión bibliografica no hubo trabajos que hablaran de brindar QoS desde la tecnología PCIe hasta una red Ethernet, consiguientemente de utilizar solo la capa 2 para poder brindar dicha QoS.

Capítulo 3

Propuesta de Marco de Trabajo de Calidad de Servicio para PCIe y Ethernet.

En este capítulo se propone un marco de trabajo que ofrece Calidad de Servicio (QoS) en una comunicación fin a fin desde PCIe hasta una red Ethernet. El marco de trabajo consta de tres partes fundamentales: control de flujo, manejo de congestión y servicios diferenciados. En cada una de éstas se describen los mecanismos que cada tecnología (PCI Express y Ethernet) van a utilizar, de igual manera se describen las relaciones que ocurren entre las clases de tráfico de cada una de ellas.

3.1. Análisis comparativo cualitativo

Este análisis consiste en comparar cuatro tecnologías de interconexión: Inifiniband (IBA), Advanced Switching (ASI), PCI Express (PCIe), y Ethernet, con el fin de observar características de cada tecnología bajo el contexto de QoS en una comunicación fin a fin.

IBA es un popular intersistema de interconexión serial, punto a punto y "full duplex" de alto rendimiento de "clusters" [15]. ASI al igual que IBA es un intersistema que provee comunicación "peer to peer", se creó con la misma tecnología de las capas de PCIe [16]. Ethernet por otra parte es la tecnología dominante para las LANs (alámbricas e inalámbricas); actualmente se utiliza para interconectar "clusters" y almacenamiento en redes de área amplia; y PCI Express a diferencia de los anteriores es un intrasistema de interconexión entre dispositivos de I/O, punto a punto y "full duplex"(Tabla 3.1) [17].

Función	IBA	ASI	Ethernet	PCI Express 1.0/2.0/3.0
Enrutamiento	"Destination lookup"	"Source routing"	"Destination lookup"	"Destination routing"
Ancho del enlace (no. de líneas)	1x, 4x, 12x	1x, 2x, 8x, 16x, 32x	1x, 4x, (para 3.125 Gb/s)	1x, 2x, 4x, 8x, 12x, 16x, 32x (en 2.0 y 3.0 llega hasta 16x)
Ancho de banda por línea	2.5, 5, 10 Gb/s	2.5, 5 Gb/s	1, 3.125, 10 Gb/s	2.5 Gb/s (1.0), 5 Gb/s (2.0), 10 Gb/s (3.0)
Ancho de banda	2.5-120 Gb/s	2.5 – 128 Gb/s	1 – 10 Gb/s	2.5 – 80 Gb/s (1.0) 5 – 80 Gb/s (2.0) 10 – 160 Gb/s (3.0)
Tamaño máximo del paquete	2096 bytes	2176 bytes	1522 bytes (9000 bytes es soportado por varios vendedores)	4113 bytes
Tamaño mínimo del paquete	24 bytes (20 bytes sin formato)	64 bytes	64 bytes	12 bytes
Codificación de la transmisión	8b/10b	8b/10b	8b/10b, 64b/66b para 10 Gb/s	8b/10b (1.0 y 2.0) 128b/130b (3.0)
Longitud máxima del cable	No Especificado	No Especificado	5000 m	No especificado
Numero máximo de hosts	49152	No Especificado	No Especificado	No Especificado
Máximos puertos por switch	255	256	No especificado	256

Tabla 3.1: Características generales de IBA, ASI, Ethernet y PCIe

Estas tecnologías de interconexión brindan Calidad de servicio (QoS) para proporcionar latencias y ancho de banda predecibles y también para darle un trato diferente a cada clase de tráfico según la finalidad de cada una de ellas.

El análisis comparativo de QoS entre las tecnologías (IBA, ASI, PCIe y Ethernet) se divide principalmente en las siguientes categorías: (1) Control de flujo, (2) Manejo de la congestión, y (3) Servicios diferenciados con la finalidad de cubrir todos los aspectos de QoS [18]. Cada tecnología utiliza diferentes términos para referirse a las mismas características de QoS (Tabla 3.2).

Termino	IBA	ASI	Ethernet	PCI Express
Enlace virtual	Linea virtual	Canal virtual	Prioridad	Canal virtual
Clase de servicio	Nivel de servicio	Clase de trafico	Clase de servicio	Clase de trafico
Paquete de la capa de enlace	Paquete	Paquete	Trama	Paquete
Interfaz de red	Adaptador de canal	"Endpoint"	Tarjeta de interfaz de red	"Endpoint"

Tabla 3.2: Terminología usada en IBA, ASI, Ethernet y PCIe

3.1.1. Control de flujo

El control de flujo tiene como objetivo eliminar o disminuir la pérdida de paquetes en la transferencia de datos. Existen diferentes técnicas para controlar el flujo, entre las más destacadas están: (1) Basado en créditos y (2) Basada en tiempo.

El mecanismo basado en créditos trabaja mediante el conteo de espacios disponibles en el "buffer" El "upstream" mantiene un registro de los créditos disponibles. Los créditos disminuyen cada que se envía un paquete. Así mismo por el lado "downstream" también tiene un registro de los créditos, al disminuir un crédito disminuye un espacio en el "buffer". El paquete no es enviado si no existen créditos disponibles, es decir, espacio necesario en el "buffer" [18]. Infiniband, Advanced Switching y PCIe son tres tecnologías que usan este mecanismo.

Por otra parte el mecanismo basado en tiempo (Xon/Xoff) define una cota superior en el "buffer" de recepción, cuando se supera esta cota el receptor envía un mensaje de OFF (Figura 3.1.b) junto con un tiempo a los emisores, que indica que se deben detener las transmisiones por el tiempo indicado en el paquete. Los emisores deben esperar el tiempo indicado, siempre y cuando no llegue otro mensaje de ON (Figura 3.1.a), que puede darse cuando el receptor ya no esté congestionado, es decir, existen espacios disponibles en el "buffer" De esta manera los emisores pueden volver a mandar paquetes [19].

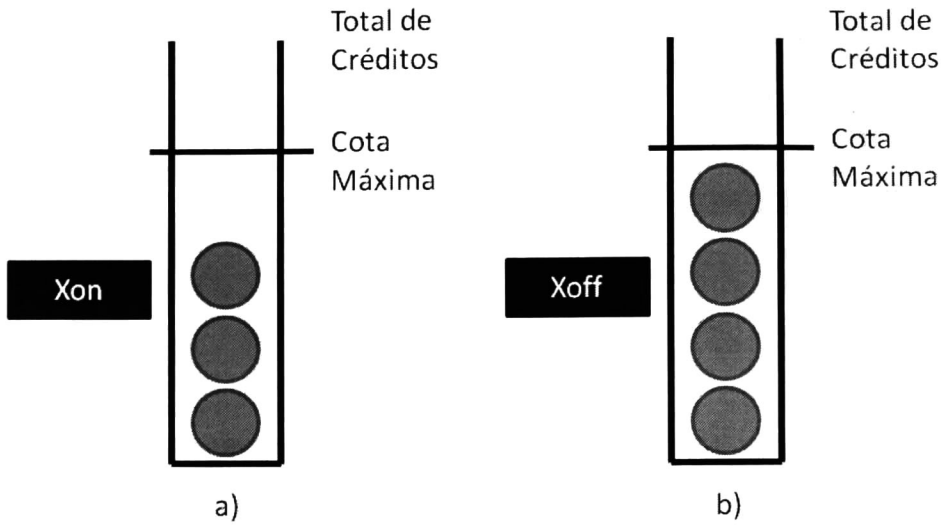


Figura 3.1: Mecanismo Xon/Xoff

Ethernet es una arquitectura que utiliza el mecanismo de control de flujo basado en tiempo (Tabla 3.3).

Termino	IBA	ASI	Ethernet	PCI Express
Control de Flujo	Basado en crédito	Basado en crédito	Xon/Xoff	Basado en crédito
Manejo de la congestión.	FECN	FECN, SBFC	ECM	Link-by-Link (Basado en créditos)
Tasa de control	Estático	(Semi) Dinámico	(Semi) Dinámico	(Semi) Dinámico
Clases de servicio	16	8	8	8
Canales virtuales	16	20	8	8
Planificador de prioridad	WFQ	Table, MinBW	SPQ, WFQ (opcional)	Round Robin, WRR

Tabla 3.3: Características de QoS de IBA, ASI, Ethernet y PCIe

3.1.2. Manejo de la congestión

El objetivo del manejo de congestión es prevenir o recuperarse ante la aparición de congestión en la red, reduciendo o controlando este efecto. IBA utiliza diferentes mecanismos para

ayudar a los conmutadores y enlaces a que no se sobrecarguen y que no se agoten sus créditos. IBA soporta 3 mecanismos diferentes: (1) "forward explicit congestion notification"(FECN), (2) "static rate control" (3) "head-of-queue drain mechanism".

IBA utiliza al mecanismo FECN para informar al destino del paquete (notificación) que está atravesando por una congestión; "static rate control" es utilizado en los nodos finales para reducir la tasa de inyección de paquetes después de recibir una notificación de congestión. Por último IBA utiliza el mecanismo "head-of-queue drain"que es el encargado de cerciorarse de que la cola de un conmutador es drenada después de un "time-out"

Por otra parte ASI utiliza otro mecanismo llamado "status-based flow-control (SBFC)", este mecanismo permite al "downstream" del conmutador cambiar de puerto de egreso si éste está congestionado. Esto es posible por que el enrutamiento es de tipo "source routing" de esta forma el paquete conoce por dónde tiene que pasar desde que sale del origen. Si el "downstream" del conmutador recibe una señal de congestión de uno de los puertos de egreso del "upstream" del conmutador, entonces permite por medio del mecanismo de planificación cambiar de puerto de egreso. Al mismo tiempo los paquetes de menor prioridad son los que usarían el puerto congestionado.

El equipo de desarrollo de Ethernet incursionó en el manejo de la congestión, hasta que se creó ECM (Ethernet Congestion Management ECM). ECM es un mecanismo que trabaja con 3 etapas: (1) Detección, (2) Señales ECM y (3) Reacción. La detección se da por medio de una cota máxima especificada en el "buffer" de un CP (Congestion point), al momento de alcanzar esa cota es cuando se detecta que la red está pasando por una congestión. Posteriormente se manda un mensaje de "slow down"(señal ECM) a los emisores, esto indica que deben de bajar la tasa de inyección de paquetes en la red (reacción). Cuando el "buffer" vuelve a tener espacio suficiente manda mensaje de "speed up"(señal ECM) lo cual indica que puede volverse a incrementar la tasa de inyección de paquetes en la red [20].

Por parte de PCIe el mecanismo de control de flujo basado en créditos ayuda a evitar la congestión en los "switches" y al definir también cotas, no se permite mandar paquetes cuando se llega a la cota máxima [3].

3.1.3. Servicios diferenciados

Los servicios diferenciados (diffserv) aplican un trato diferente al tráfico de acuerdo a las garantías que necesita cada tipo tráfico. IBA soporta líneas (canales) virtuales y logra la diferenciación de tráfico ya que cada nivel de servicio es asignado a cada línea virtual, este

nivel puede ser de alta o baja prioridad. IBA soporta un máximo de 16 líneas virtuales y 16 niveles de servicio. Por otra parte ASI soporta "diffserv" usando clases de tráfico que trabajan con flujos independientes unos de otros. ASI utiliza 8 clases de tráfico que son asignadas a 20 canales virtuales permitiendo la utilización de diferentes prioridades dependiendo el canal virtual (Tabla 3.3).

Ethernet aplica "diffserv" basándose en prioridades e implementando múltiples colas, utiliza 8 diferentes niveles de prioridad y utiliza "Strict Priority Queuing"(SPQ), aunque también soporta "Weighted Fair Queuing" (WFQ). El algoritmo de SQP trabaja con la política FIFO y tiene problemas al tratar con la menor prioridad ya que solo se atienden paquetes de esta prioridad cuando no existan paquetes de las otras prioridades esperando a ser atendidos. Una solución dada a este problema fue el algoritmo WFQ que trabaja bajo un esquema "round robin" dándole un peso diferente a cada nivel de prioridad y atendiéndolos de acuerdo a ese peso, de esta manera se atienden de mejor forma los paquetes de más baja prioridad [21].

PCIe soporta servicios diferenciados y utiliza 8 clases de tráfico y 8 canales virtuales. Para arbitrar estas clases de tráfico utiliza dos métodos llamados "port arbitration y VC arbitration" cada uno de estos métodos utiliza diferentes mecanismos de encolamiento como: (1) "Hardware-fixed arbitration" (2)"Weighted Round Robin(WRR)" y (3) "Time-Based WRR" [3].

El algoritmo "Hardware-fixed arbitration" es basado en "round robin" donde a cada puerto le da la misma prioridad. WRR trabaja similar al algoritmo WFQ de Ethernet, la única diferencia es la asignación de diferentes valores de fases en lugar de pesos. En cambio el "Time-Based WRR" es requerido para las transacciónes isócronas de PCIe.

Al recopilar información y el análisis posterior de la misma, se encontraron puntos específicos para dar origen a la propuesta descrita en la siguiente sección.

3.2. Descripción

Se propone un marco de trabajo (Figura 3.2) que brinda QoS en una comunicación fin a fin desde PCI Express hasta una red Ethernet. Éste está dado por un enfoque de incorporación y soporte de la parte lógica y de mecanismos relacionados a la QoS de PCIe en un conmutador Ethernet. Éstos incluyen diferentes mecanismos de control de flujo (CF) y manejo de la congestión (CM) con la finalidad de disminuir latencias en los paquetes y aumentar el rendimiento en la red.

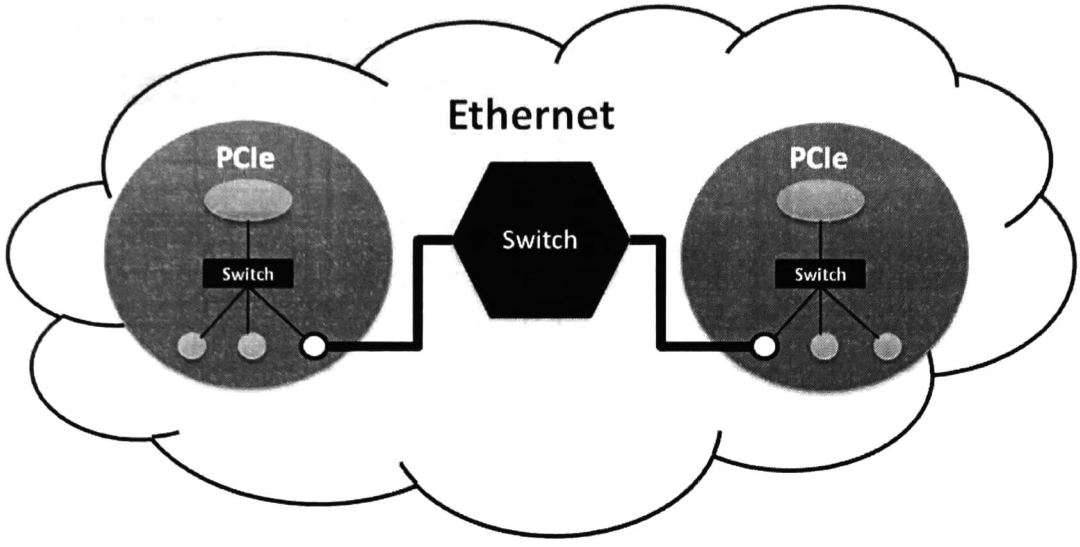


Figura 3.2: Comunicación fin a fin desde PCIe hasta Ethernet

El marco de trabajo (Figura 3.3) es definido con el nombre de *Framework* y está conformado por varias partes:

$$Framework = \{(CF), (CM), (Ent), (Rel)\}$$

donde: CF se compone por dos mecanismos diferentes de control de flujo que utilizan Ethernet (Xon/Xoff) y PCIe (basado en créditos). CM es el conjunto de mecanismos de manejo de congestión que utilizan Ethernet (ECM) y PCIe (Link-by-Link). *Ent* es el conjunto de entidades necesarias para poder brindar marcado, diferenciación de paquetes y arbitraje para que Ethernet y PCIe puedan trabajar conjuntamente. *Rel* es un conjunto de relaciones definidas para cada paquete de PCIe y Ethernet, de tal forma que a dicho paquete le corresponda un determinado canal virtual. De esta forma se dismunnuye el número de veces en que los paquetes son etiquetados.

3.3. Control de flujo

Se analizaron varios mecanismos de control de flujo para controlar y disminuir la pérdida de paquetes. Como resultado de dicho análisis, en la tecnología de PCIe se optó por seguir utilizando el mecanismo de control de flujo basado en créditos (Figura 3.4), visto que, la

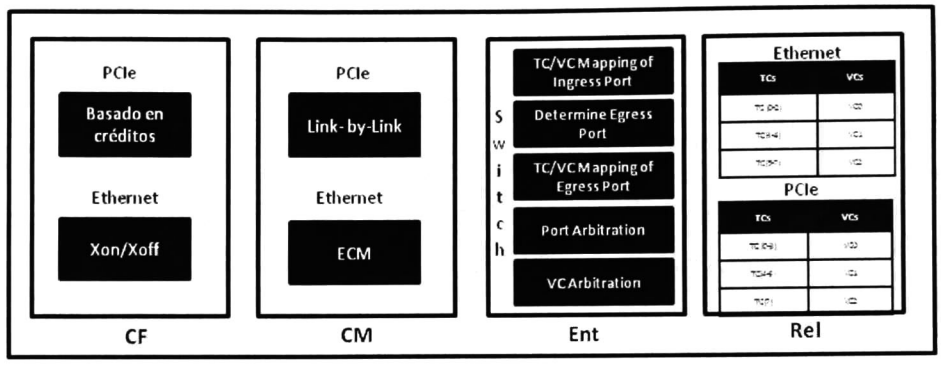


Figura 3.3: Framework

comunicación de PCIe es punto a punto, es necesario un mecanismo que asegure que los paquetes siempre llegan a su destino. El mecanismo basado en créditos se adapta de manera satisfactoria, ya que no se puede enviar un paquete si no existen créditos suficientes para que dicho paquete sea enviado. De esta manera los paquetes no se pierden debido a que el "buffer" notifica cuando está lleno, lo que implica que no tiene créditos y no puede recibir más paquetes.

En la tecnología Ethernet se utilizará el mecanismo Xon/Xoff, puesto que la conectividad es muchos a uno, este mecanismo se adecua mejor a este tipo de conexiones. Xon/Xoff manda el mensaje de "Xoff" a los emisores cuando se sobrepase la cota máxima junto con un tiempo determinado. Esta cota máxima es menor al número de créditos disponibles en el "buffer" receptor. Posteriormente los emisores esperan el tiempo indicado en el mensaje, para volver a mandar un paquete, siempre y cuando no hayan recibido un mensaje de "Xon". Este mecanismo junto con el mecanismo de manejo de la congestión "Ethernet Congestion Management" (ECM) ayudan en gran medida a evitar la pérdida de paquetes (Figura 3.5).

3.4. Manejo de la congestión

El manejo de congestión tiene el propósito de evitar que se saturen los conmutadores y prevenir que se acaben los créditos disponibles para que los emisores puedan mandar paquetes a los receptores.

En el marco de trabajo, PCIe manejará un mecanismo de control de flujo "Link-by-link"

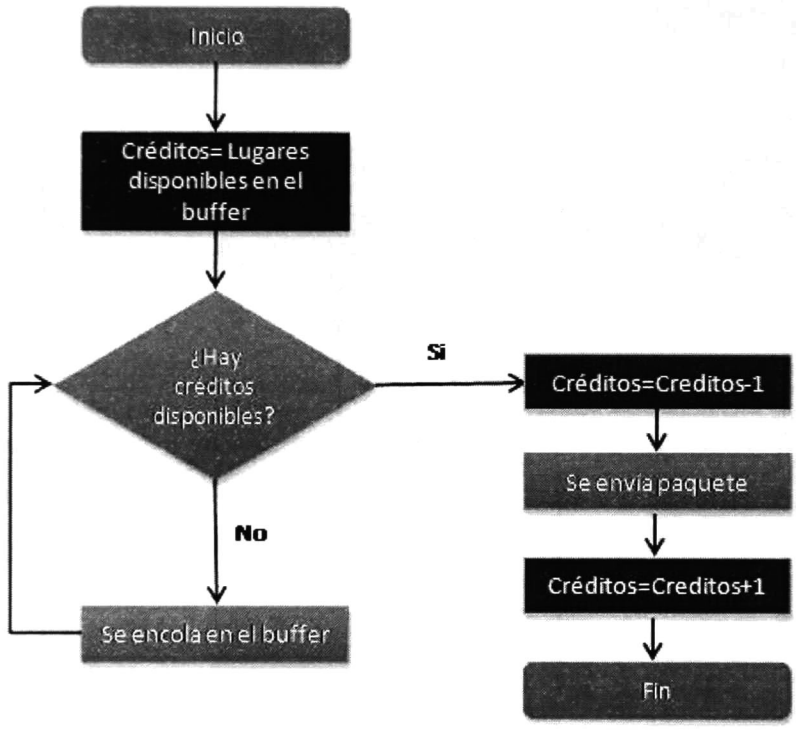


Figura 3.4: Diagrama de flujo del mecanismo basado en creditos usado por PCIe

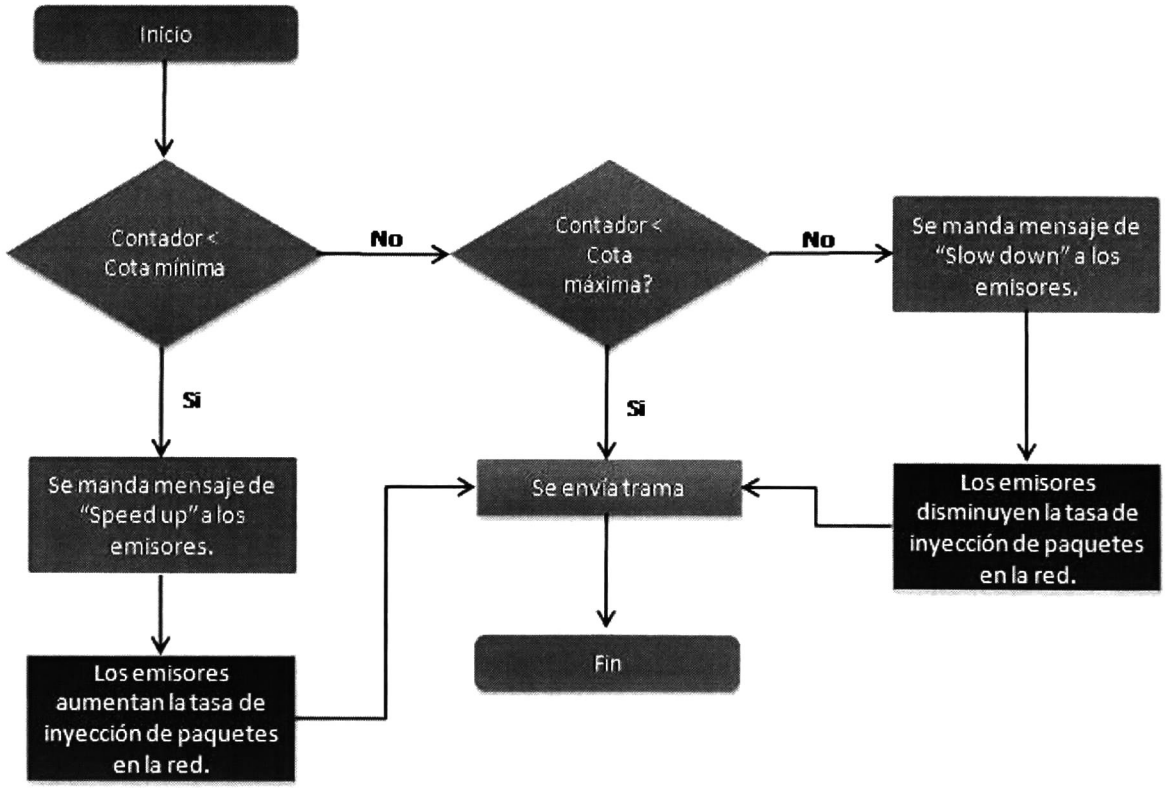


Figura 3.5: Diagrama de flujo del mecanismo ECM usado por Ethernet

dado que la categoría de control de flujo utiliza el mecanismo basado en créditos, éste a su vez sirve para manejar la congestión que se da por cada enlace, si no hay créditos disponibles no se puede mandar un paquete hasta que existan créditos suficientes para el paquete.

Con respecto a Ethernet se trabajará con el mecanismo ECM. ECM se basa en varias etapas: (1) "Detection", (2)"ECM signals" y (3) "Reaction" El mecanismo ECM define una cota máxima en el "buffer" del receptor, esta cota es menor a los créditos totales con los que cuenta el "buffer" Cuando se rebasa esta cota máxima, se dice que hay una congestión y de esta forma es detectada.

Después de haber detectado una congestión, el punto de congestión manda un "ECM signal" a cada uno de los emisores con un mensaje de "slow down" indicando que la red está pasando por una congestión. Los emisores deben reaccionar ante este mensaje disminuyendo la tasa de inyección de tráfico en la red. Cuando el punto de congestión se restablece, es decir, ya no se encuentra congestionado, manda de nuevo un "ECM signal" con un mensaje de "speed up" a cada uno de los emisores indicando que pueden aumentar de nuevo su tasa de inyección de datos.

Posteriormente a que el paquete es recibido en el "buffer", éste se atenderá por un mecanismo de "round robin"

3.5. Servicios diferenciados

Es muy importante poder distinguir los paquetes para darles un trato diferente, de acuerdo a la prioridad de cada uno. Normalmente la clasificación de paquetes se da en una red homogénea, en nuestro caso lo hacemos en una red heterogénea (Ethernet y PCIe).

De acuerdo al marco de trabajo propuesto, QoS se brinda solo sobre la capa 2 en las dos tecnologías (Ethernet y PCIe), sin la necesidad de subir a otras capas. De modo que no es necesario agregar más datos (QoS en capa 3) en la trama, por lo que dichas tramas se verán beneficiadas en métricas como latencia, rendimiento y perdida de paquetes (Figura 3.6).

Para brindar QoS en la capa 2 en la tecnología Ethernet, se agregarán campos (id y tc) necesarios para poder diferenciar e identificar las tramas (Figura 3.7). EL campo ïd"se agregará con la finalidad de llevar un control de la trama y el campo "tc"para asignarle una determinada clase de tráfico a la trama y poder hacer la diferenciación de tramas, dado que, a cada clase de tráfico le corresponde un canal virtual, cada uno con diferente prioridad.

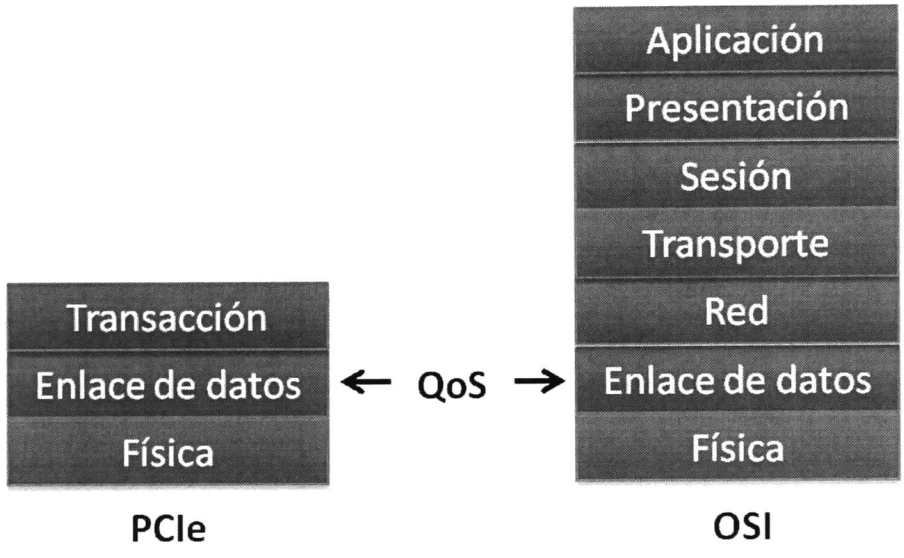


Figura 3.6: Brindando QoS en la capa de enlace de datos.

Figura 3.7: Agragación de campos en la cabecera de la trama

Para lograr la clasificación de paquetes se aplica un mecanismo en el conmutador. Este mecanismo utiliza varias entidades que ayudan a diferenciar los paquetes y darles prioridades.

Se propone utilizar parte de la lógica del conmutador de PCIe en la tecnología Ethernet, de esta forma se podrán identificar paquetes provenientes de PCIe y relacionarlos inmediatamente sin la necesidad de volver a marcar cada paquete. Ésta lógica se compone por varias entidades (Figura 3.8) que serán las encargadas de proporcionar diferenciación y arbitraje en los paquetes. Así mismo se utilizará una entidad llamada adaptador que nos ayudará a relacionar y asignar canales virtuales a los tipos de clases de cada paquete.

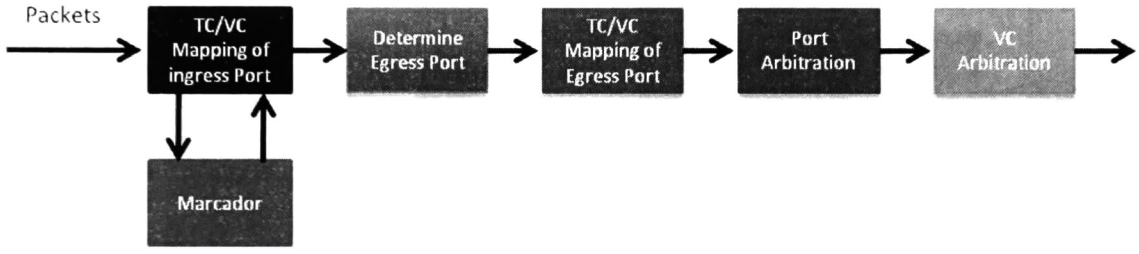


Figura 3.8: Entidades utilizadas con la finalidad de brindar diferencación de paquetes.

El conmutador utiliza diferentes entidades para poder brindar diferenciación de tramas y etiquetado (Figura 3.9) las cuales se describen a continuación:

1. "TC/VC Mapping of Ingress Port"

El "TC/VC Mapping of Ingress Port" realiza varias funciones, donde la principal es la de lograr establecer una relación entre los paquetes de PCIe y las tramas Ethernet, con la finalidad de marcar las tramas una sola vez y no tener la necesidad de volver a marcarlas. Primero el asignador verifica si la trama tiene asignado un VC, de no ser así, marca la trama y le asigna un VC. De esta manera las tramas llegan directamente a su respectivo "buffer" de acuerdo al VC al que pertenecen.

2. "Determine Egress Port"

El "Determine Egress Port" encamina la trama al puerto de egreso correspondiente de acuerdo a la información de ruteo de cada trama.

3. "TC/VC Mapping of Egress Port"

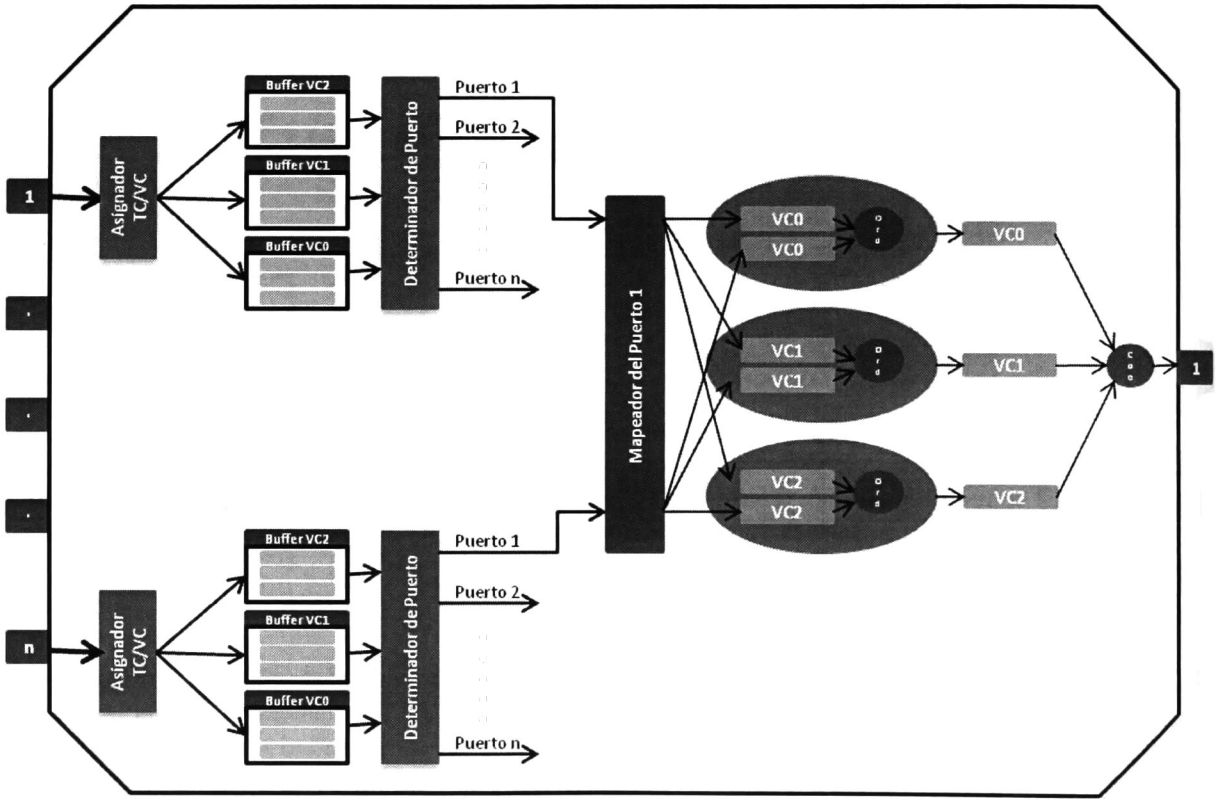


Figura 3.9: Arquitectura de los switch

Por cada puerto de egreso, existe un "TC/VC Mapping of Egress Port" encargado de organizar las tramas provenientes de diferentes puertos de entrada en los VCs al que corresponden de acuerdo a su identificador (ID).

4. "Port Arbitration"

El "Port Arbitration" se encarga de ordenar las tramas de cada VC y darles salida a las tramas de cada uno de ellos.

5. "VC Arbitration"

El "VC Arbitration" se encarga de darle salida a las tramas de acuerdo a las reglas de prioridades de los VCs.

El conmutador utiliza las entidades previamente descritas (Figura 3.10) para poder brindar diferenciación de tramas y relaciones que se requieren entre Ethernet y PCIe.

1. *Se Verifica si el paquete ya tiene asignado su correspondiente VC.*
   Si VC= Vacio
   Se Asigna el VC de acuerdo a su TC.
2. *Se encola el paquete en el buffer correspondiente de acuerdo a su VC.*
3. *Se determina el puerto de salida del paquete y se encola en el buffer general de su VC.*
4. *El arbitro "Arb" va atendiendo paquetes de acuerdo a un mecanismo de Round Robin (RR).*
5. *El arbitro "Coo" atiende los paquetes de acuerdo a las prioridades de los VCs y aplica un mecanismo de "Weight Round Robin"*

Figura 3.10: Funcionamiento del conmutador

El mecanismo "Weight Round Robin" (WRR) utiliza diferentes pesos de acuerdo a la prioridad de cada uno de los VCs. En nuestro caso solo se utilizan 3 VCs (VC0, VC1, VC2) siendo VC2 el de mayor prioridad y al que se le asigna más peso, es decir se procesan más paquetes de este tipo que de los VCs de menor prioridad (VC0) (Figura 3.11).

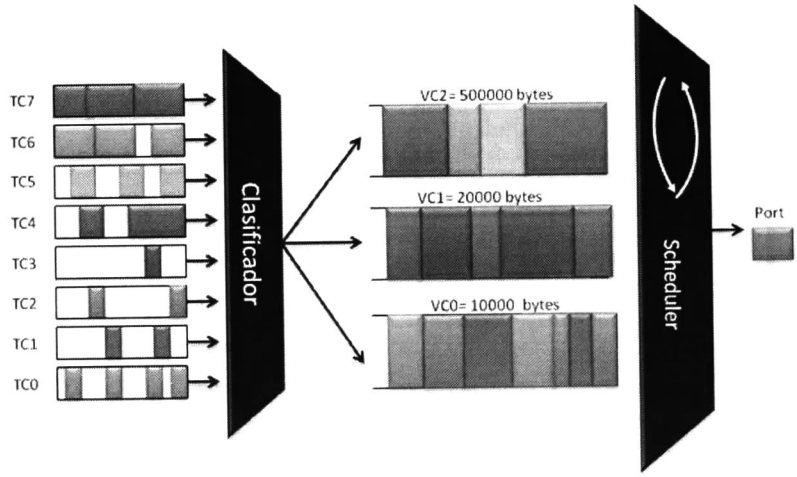


Figura 3.11: Ejemplo de WRR en un conmutador Ethernet con 3 VCs y sus respectivos pesos

Capítulo 4

Simulacion del marco de trabajo

En esta sección se muestra una descripción, diseño y código utilizado para desarrollar un simulador del marco de trabajo propuesto. Este simulador se utilizó para realizar pruebas generales a efecto de demostrar que los mecanismos propuestos mejoran el rendimiento de la red. Éstas se describen en el próximo capitulo.

4.1. Especificación

El simulador del marco de trabajo se desarrolló por la necesidad de probar que las ideas antes descritas realmente funcionarán. A efecto de probar que con la extensión en la cabecera de las tramas Ethernet se puede disminuir las veces en las que se marca un paquete y por tanto brindar QoS desde la capa 2.

4.1.1. Función del simulador

La principal función del simulador recae en extender las capacidades de QoS en el dominio PCIe-Ethernet. Esto se obtiene por medio de la ampliación de la cabecera de la trama, agregándole campos necesarios para la diferenciación e identificación de tráfico y para relacionar paquetes PCIe con tramas Ethernet con la finalidad de disminuir las veces en que se marcan los paquetes.

4.1.2. Objetivos del Simulador

- OBJ-01 Incorporación del conmutador: Incorporación de la lógica del conmutador de la tecnología PCIe al conmutador de la tecnología Ethernet bajo el contexto de QoS ("diffserv").
- OBJ-02 Extension de la cabecera: Extensión de la cabecera (campos) de la trama de la tecnología Ethernet para el soporte de QoS.
- OBJ-03 Transmisión de datos: Transmisión de tramas Ethernet con la cabecera extendida por medio de la incorporación de la lógica del conmutador (PCIe) al ambiente Ethernet.

4.1.3. Requerimientos

- RS-01 "Modulo Conmutador"
  - Función: Brindar QoS en capa 2.
  - Descripción: El propósito de este requisito es brindar QoS ("diffserv") por medio de la adaptación de la lógica utilizada en el comnutador de PCIe en el conmutador Ethernet (capa 2).
  - Precondición: La trama ya debe de contar con la cabecera extendida.

  RS-02 "Mapping TC-VC"

- Función: Marcar paquetes y asignarlos a sus recpectivos VCs.
- Descripción: El propósito de este requisito es verificar si un paquete ya ha sido marcado, si no, éste es marcado y asignado a su correspondiente VC.
- Precondición: La trama ya debe de contar con la cabecera extendida.

- RS-03 "Port Arbitration"
  - Función: Coordinar tramas pertenecientes al mismo VC.
  - Descripción: El propósito de este requisito es dar salida a paquetes que están colocados en una cola tipo FIFO.
  - Precondición: La trama ya debió haber sido asignada a su correspondiente VC.

  RS-04 "VC Arbitration"
  - Función: Coordinar paquetes pertenecientes a diferentes VCs.
  - Descripción: El propósito de este requisito es dar salida a tramas que están colocadas en diferentes VCs, por medio de un mecanismo WRR.
  - Precondición: La trama ya debió haber sido asignada a su correspondiente VC.
- RC-01 "Agregación de campos"
  - Función: Extender la cabecera de las tramas Ethernet.
  - Descripción: El propósito de este requisito es extender la cabecera de las tramas Ethernet por medio de la agregación de campos necesarios para poder diferenciar y priorizar tramas.

  RER-01 "Enviar/Recibir Tramas"
  - Función: Enviar y recibir tramas Ethernet.
  - Descripción: El propósito de este requisito es enviar y recibir tramas de la tecnología Ethernet brindando QoS ("diffserv") desde la capa 2.
  - Precondición: La trama ya debe de contar con la cabecera extendida.

4.1.4. Apendice

- OBJ: Abreviatura de objetivo.
- RS: Abreviatura de requisitos del conmutador.
- RC: Abreviatura de requisitos de la cabecera.
- RER: Abreviatura de requisitos del modulo enviar-recibir.

4.2. Diseño

El simulador se compone de tres partes basicas (Figura 4.1) para extender la QoS desde la tecnología PCIe hasta la tecnología Ethernet:

- Conmutador
  Cabecera
- Modulo enviar-recibir

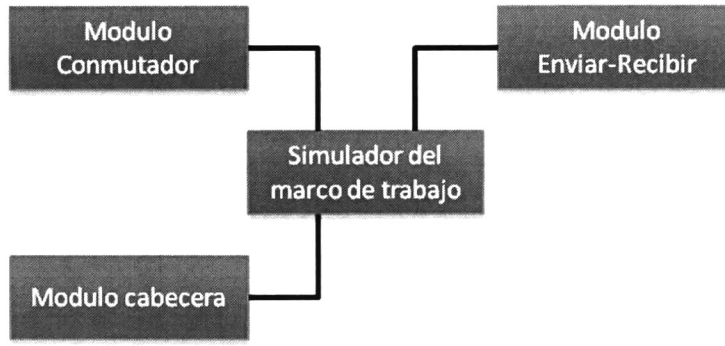


Figura 4.1: Diagrama de composición modular del simulador del marco de trabajo

4.2.1. Conmutador

Se creó una clase llamada "switch" la cual se basa en la lógica utilizada en la tecnología PCIe para poder diferenciar paquetes. Ésta se adaptó a la tecnología Etehernet, logrando brindar y diferenciar tramas desde la capa 2 (enlace de datos), sin la necesidad de brindarla en capa 3 (red).

La lógica del conmutador de PCIe trabaja con 5 entidades (Figura 4.2) que realizan diferentes actividades que nos ayudan a brindar QoS ("diffserv"), marcado de paquetes y priorización de VCs.

Cada una de estas entidades sigue un determinado orden para poder atender a los paquetes de una manera adecuada. Primero el paquete es atendido por la entidad "TC/VC Mapping of

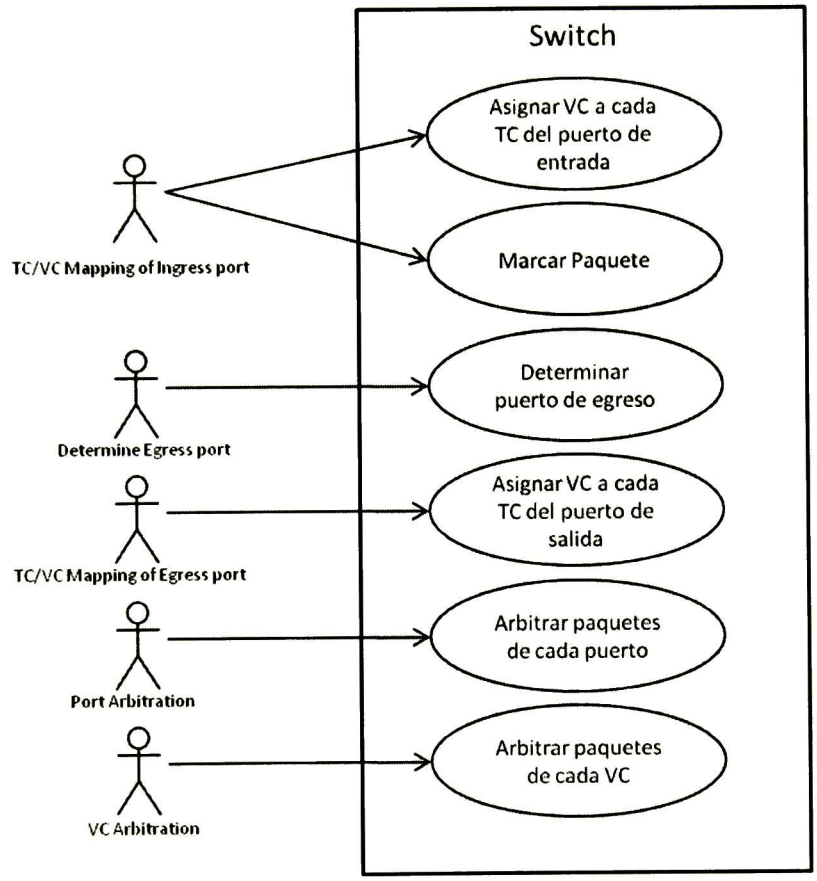


Figura 4.2: Diagrama de casos de uso del conmutador

Ingress Port" que será la encargada de verificar si el paquete ya fue marcado, si no es así, ésta marcará el paquete y lo asignará a su correspondiente VC. Posteriormente se determinará el puerto por donde saldrá el paquete ("Determine Egress Port"). En consecuencia el paquete es de nuevo asignado a su VC ahora de cada puerto de salida ("TC/VC Mapping of Egress Port"), cada paquete de cada cola de cada VC es atendido por medio de un mecanismo FIFO ("Port Arbitration"). Por último cada paquete es atendido de acuerdo a la prioridad de su VC, por medio de un WRR ("VC Arbitration").

Para un estudio general del conmutador se redujo a tener un solo puerto de entrada y uno de salida por lo que las entidades básicas serían: (1) "TC/VC Mapping of Ingress Port" (2) "Port Arbitration" y (3) "VC Arbitration" (Figura 4.3).

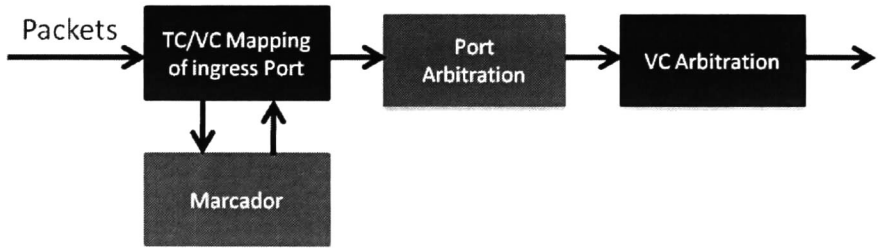


Figura 4.3: Diagrama de flujo del conmutador con un solo puerto de entrada y un solo puerto de salida.

4.2.2. Cabecera

Para poder utilizar la clase "switch" fue necesario realizar ciertas modificaciones a la cabecera de las tramas Ethernet. A ésta se le agregaron campos necesarios (id y tc) para poder diferenciar e identificar tráfico con el fin de brindar QoS desde la capa 2 y para disminuir las veces en que se marcan los paquetes ya que se da una relación entre los paquetes de PCIe y las tramas Ethernet (Figura 4.4).

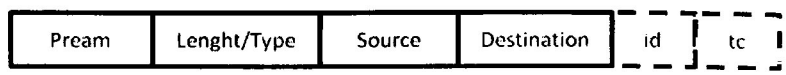


Figura 4.4: Trama con los campos id y tc agregados.

4.2.3. Modulo enviar-recibir

Para poder transmitir datos de un nodo a otro en NS-3 utilizando la clase creada "switch" se utilizó el modulo Openflow (Figura 4.5). Éste modulo cuenta con conmutadores. Éstos conmutadores son configurados por medio de la Interfaz de programación de aplicaciones ("Application Programming Interface" API) "OpenFlow" y también tienen una extensión MPLS (MultiProtocol Label Switching) para el soporte de QoS.

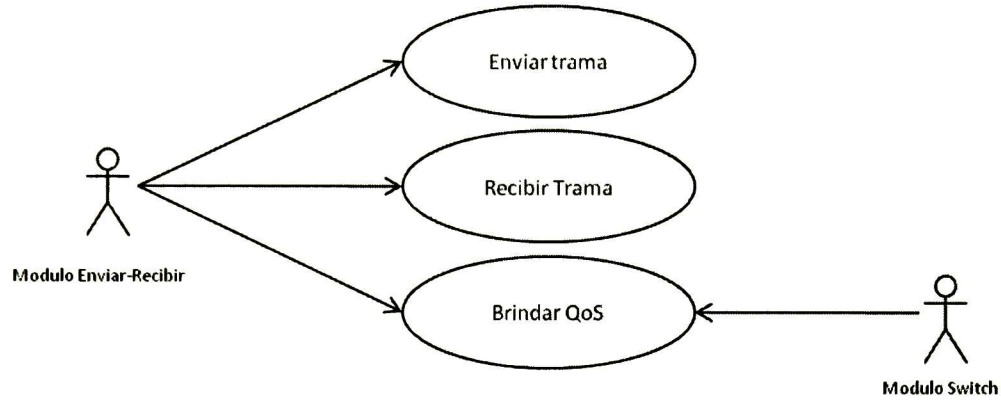


Figura 4.5: Diagrama de casos de uso del modulo enviar-recibir.

El módulo de "OpenFlow" presenta un "OpenFlowDevice" y un "OpenFlowSwitchHelper" para ser instalados sobre los nodos. Semejante al módulo del puente (bridge), toma una colección de "NetDevice" para establecerlos como puertos, y actúan como intermediarios entre ellos, recibiendo un paquete sobre un puerto y mandándolo sobre otro o sobre todos.

Como un conmutador "OpenFlow" mantiene una tabla configurable de flujo que puede relacionar paquetes mediante sus cabeceras y realizar diferentes acciones con el paquete basados sobre cómo este los relaciona.

La funcionalidad se pasa al controlador, el cual envía mensajes al conmutador que configura su flujo, produciendo diferentes efectos. Los controladores pueden ser agregados por el usuario, bajo el nombre de espacio "ofi" extendiendo "ofi::Controller" Usuario con conocimientos del estándar "OSFID" y/o del protocolo "OF "puede crear controladores virtuales para crear conmutadores de diferentes tipos de clases.

Como este módulo nos permite la agregación de funciones lógicas, se optó por usarlo y

agregarle la funcionalidad de la clase "switch" para brindar QoS desde la capa 2.

4.3. Desarrollo

Para el desarrollo del simulador del marco de trabajo se utilizó el Simulador de Redes 3 ("Network Simulator 3 NS-3"), con el fin de utilizar clases existentes para modificarlas y poder cumplir con nuestros objetivos.

NS-3 es un simulador de eventos discretos de la red de sistemas de Internet, dirigidos principalmente a la investigación y uso educativo. NS-3 es un software libre, licenciado bajo la licencia GNU GPLv2, y está a disposición del público de investigación, desarrollo y uso.

NS-3 está diseñado como un posible reemplazo para el popular simulador NS-2. El nombre de dominio siglas "nsnam"deriva históricamente de la concatenación de ns (simulador de red) y nam (animador de red).

NS-3 responde a las tendencias de cómo la investigación de Internet se lleva a cabo:

- Base de software extensible
- Atención al realismo
- Software de integración
- Apoyo para la virtualización y bancos de pruebas
- Flexible de seguimiento y estadísticas
- Atributo del sistema
- Nuevos modelos

Por otra parte se utilizó la clase "Ethernet-header" de NS-3 y se le hicieron modificaciones para que contara con los campos "tc" e "id" para hacer la diferenciación de tramas y asignar un canal virtual según su prioridad.

El algoritmo del simulador del marco de trabajo funciona de la siguiente manera:

1. El conmutador recibe el paquete.
2. Se obtiene la cabecera Ethernet, y verifica el tipo de clase de la trama.
3. Se almacena la trama dependiendo el tipo de clase, en el canal virtual correspondiente "*FlowTableLookup*" indicándole el "id" de la trama y el puerto de salida.

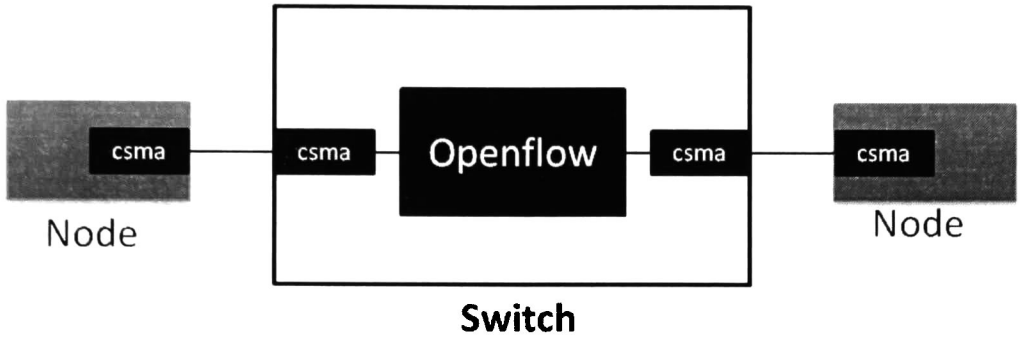


Figura 4.6: Arquitectura del simulador del marco de trabajo.

4. La función "*RunThroughFlowTable*" manda llamar la función "*FlowTableLookup*" con un cierto retardo (entre 0.03755 y 3.89544 mseg).
5. La función "*FlowTableLookup*" encola todos las tramas que se quieren enviar y también encola los datos necesarios para que la trama sea enviada en caso de que el canal esté ocupado y retransmitirlo en otro tiempo. También llama a la función "*Transmiting*()", la cual indica que llego una trama y se quiere enviar.
6. La función "*Transmiting*()" :

   Verifica que el canal esté libre, si no lo está se sale de la función.

   - Si está libre entonces obtiene la siguiente trama a transmitir usando la lógica del conmutador de PCIe para la selección de la siguiente trama, la función utilizada es "*arbitrator*()"
   - Después obtenemos los datos correspondientes de la trama obtenida que son necesarios para el envío de ésta.

     Ya obtenidos las tramas se ejecuta la acción de enviar el trama.
   - Se cambia el estado del canal a ocupado.

     Se asigna un tiempo dependiendo del tamaño del "buffer" de la trama para que el canal se libere utilizando la función "*libera*()" y "*Simulator* :: *Schedule*(..)" la cual da el retardo para la ejecución de la función libera "()"
7. La función libera(*uint32_upacket_uid*) se ejecuta pasando el tiempo de simulación, el cual libera el canal y da permiso para transmitir otro paquete.

Capítulo 5

Resultados y Análisis

En este capítulo se presenta una serie de simulaciones y análisis de casos de estudios, que nos permiten observar los resultados obtenidos al implementar nuestro marco de trabajo de comunicación fin a fin desde PCI Express hasta Ethernet.

5.1. Caso de estudio 1

Este caso de estudio busca demostrar el funcionamiento normal de un conmutador Ethernet, en el cual no se brinda QoS (no aplica diferenciación de tramas). En éste las tramas viajan por un VC y son encolados en un solo "buffer"

Para realizar esta simuilación se necesitó el siguiente equipo:

Hardware: PC compatible con procesador Intel x86 (x86 64) o compatible.

- Software:
  - **Sistema Operativo:** Sistema operativo GNU/Linux.
  - **NS-3:**
    - **NS-3:** Código fuente de NS-3.
    - **C++:** gcc, g++, python.
    - **Python bindings (opcional):** python-dev.
    - **Trabajo con repositorio:** mercurial.
    - **Depuración (opcional):** gdb, valgrind.
    - **Manipulación de archivos Pcap (opcional):** tcpdump.
    - **Estilo de codificación:** uncrustify.
    - **Generación de documentación:** doxygen, imagemagick, graphviz.
    - **Visualizador de simulación (opcional):** python-graphviz, python-wiki, python-pygoocanvas, libgoocanvas.

Se hizo una simulación compuesta por dos nodos y un conmutador. Ésta trabaja con conmutadores Ethernet. Cada paquete es tratado de igual manera y se atiende de acuerdo a una cola "fifo" es decir, el primero que llega al "buffer" es el primero en ser atendido.

Se creó la red en el simulador NS-3 (Figura 5.1) de acuerdo al diseño previamente descrito. Ésta red esta Compuesta por 3 nodos, en la cual los nodos que están en los extremos son los nodos finales y el nodo que esta en medio es el nodo que trabaja como conmutador Ethernet.

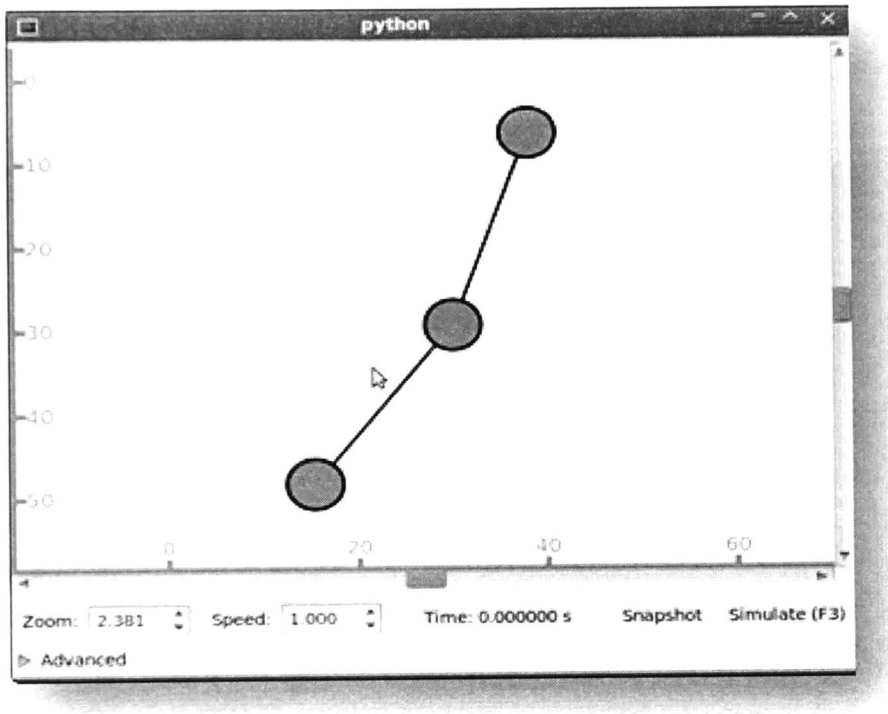


Figura 5.1: Simulación de la red del caso de estudio 1 en NS-3.

Se generaron 300 paquetes y se enviaron de un nodo a otro atravesando por el conmutador donde no se brinda QoS (Figura 5.2), es decir, todos los paquetes se atienden de igual manera.

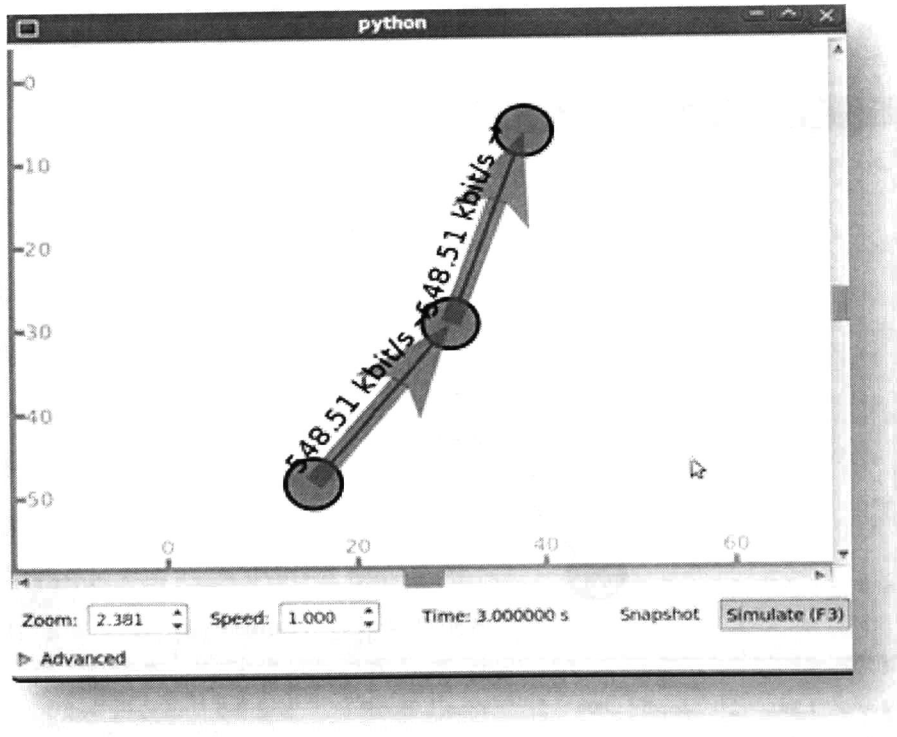


Figura 5.2: Simulación de la transmision de datos del caso de estudio 1 en NS-3.

Características de la simulación del caso de estudio 1:

- **Paquetes:** 300.
- **Tiempo de simulación:** 5 segundos.
- **Número de Nodos:** 2.
- **Tipo de canal:** CSMA.

Número de conmutadores: 1 (Ethernet)

Número de TCs: No aplica.

Número de VCs: No aplica.

Al correr la simulación se obtuvieron latencias que variaban desde 0.03755 hasta 3.89544 milisegundos (mseg) con una media de 1.620297 mseg (Figura 5.3).

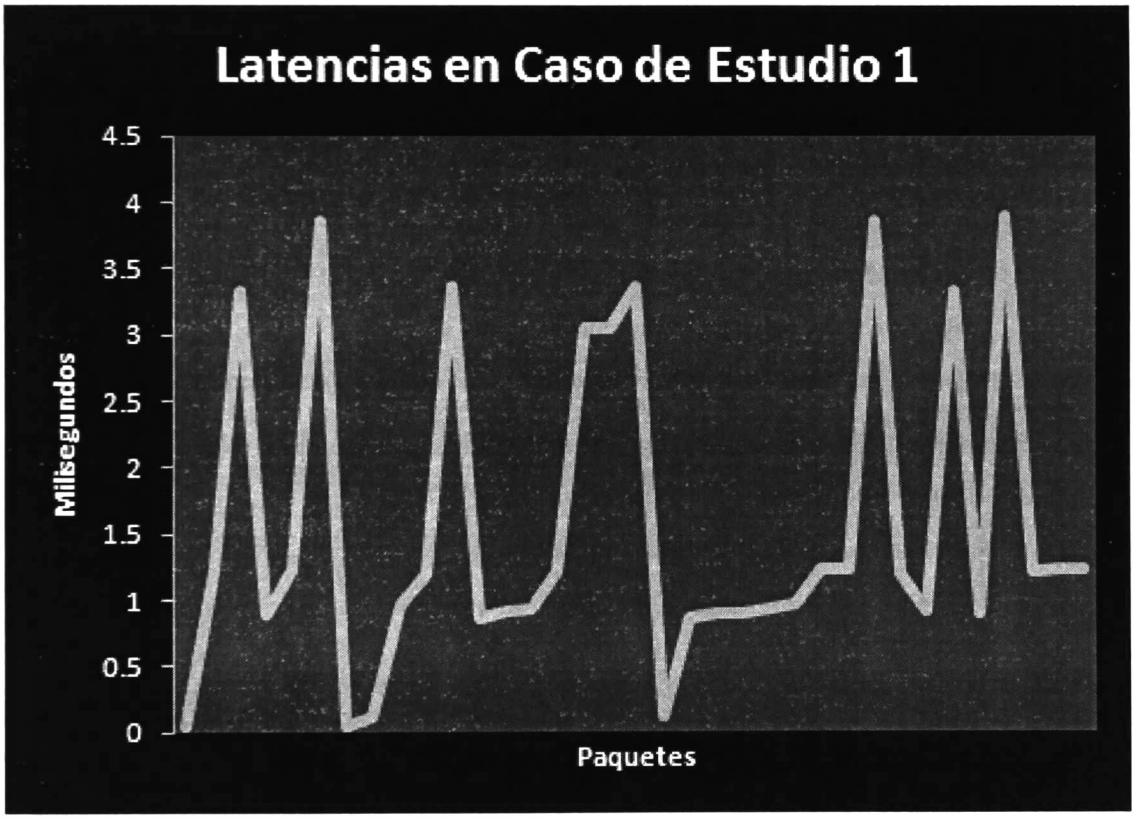


Figura 5.3: Tiempos de latencia del caso de estudio 1.

5.2. Caso de estudio 2

Este caso de estudio busca demostrar los conceptos que rigen y dan forma al marco de trabajo propuesto, tales como, el rendimiento, la latencia, paquetes y reconocimiento de tráfico.

Para realizar esta simuilación se necesitó el siguiente equipo:

Hardware: PC compatible con procesador Intel x86 (x86 64) o compatible.

Software:

- **Sistema Operativo:** Sistema operativo GNU/Linux.
- **NS-3:**
  - **NS-3:** Código fuente de NS-3.
  - **C++:** gcc, g++, python.
  - **Python bindings (opcional):** python-dev.
  - **Trabajo con repositorio:** mercurial.
  - **Depuración (opcional):** gdb, valgrind.
  - **Manipulación de archivos Pcap (opcional):** tcpdump.
  - **Estilo de codificación:** uncrustify.
  - **Generación de documentación:** doxygen, imagemagick, graphviz.
  - **Visualizador de simulación (opcional):** python-graphviz, python-wiki, python-pygoocanvas, libgoocanvas.

Se hizo una simulación en forma general conformada por una red Ethernet, la cual esta compuesta por dos nodos y un conmutador (Figura 5.3). Esta red trabaja con un conmutador que implementa la lógica de los conmutadores de la tecnología PCIe para poder brindar QoS desde la capa 2 (enlace de datos) por medio de la extensión de la cabecera.

Cada paquete enviado por el nodo pasa por el Asignador TC/VC, el cual verifica si el paquete está etiquetado, si no, lo etiqueta. Como sólo se cuenta con un puerto de salida pasa

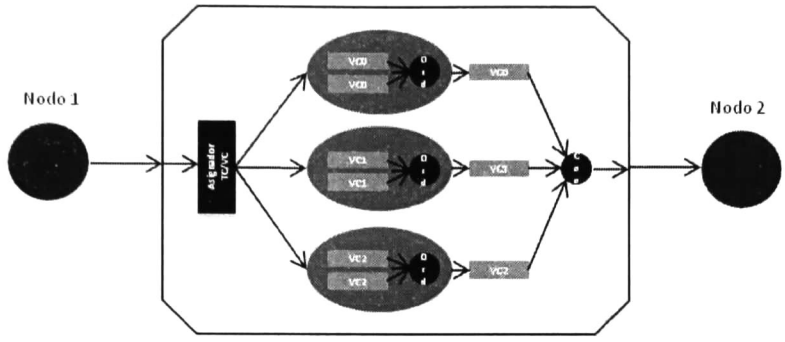


Figura 5.4: Simulación de una red Ethernet, trabajando con la lógica de un conmutador PCIe.

directamente a encolarse a su correspondiente VC según su TC. Posteriormente el ordenador dirige el paquete al coordinador por medio de un "Round Robin" El coordinador por último envía cada paquete al nodo correspondiente dependiendo su prioridad de acuerdo a un WRR.

Así mismo se utiliza la verificación de marcado para disminuir las veces en que se marcan las tramas.

Se creó la red en el simulador NS-3 (Figura 5.4) de acuerdo al diseño descrito previamente. Dicha red esta compuesta por 3 nodos, en el cual los nodos que están en los extremos son los nodos finales y el nodo del medio trabaja como conmutador por medio del módulo "Openflow"

Se generaron 300 paquetes con un TC aleatorio y se enviaron de un nodo a otro atravesando por el conmutador donde se les brinda QoS (Figura 4.3). Al correr la simulación se obtuvieron diferentes datos que mostraron que el tiempo de transmision del VC2 (mayor prioridad) fue menor que los tiempos de transmisión de los VC1 y VC0, por tanto la diferenciación de paquetes trabajó adecuadamente y se pudo brindar QoS desde la capa 2 (enlace de datos) sin tener la necesidad de brindarla en la capa 3 (red) (Figura 5.4).

Características de la simulación del caso de estudio 2:

Paquetes: 300 con TC aleatorio.

- **Tiempo de simulación:** 5 segundos.

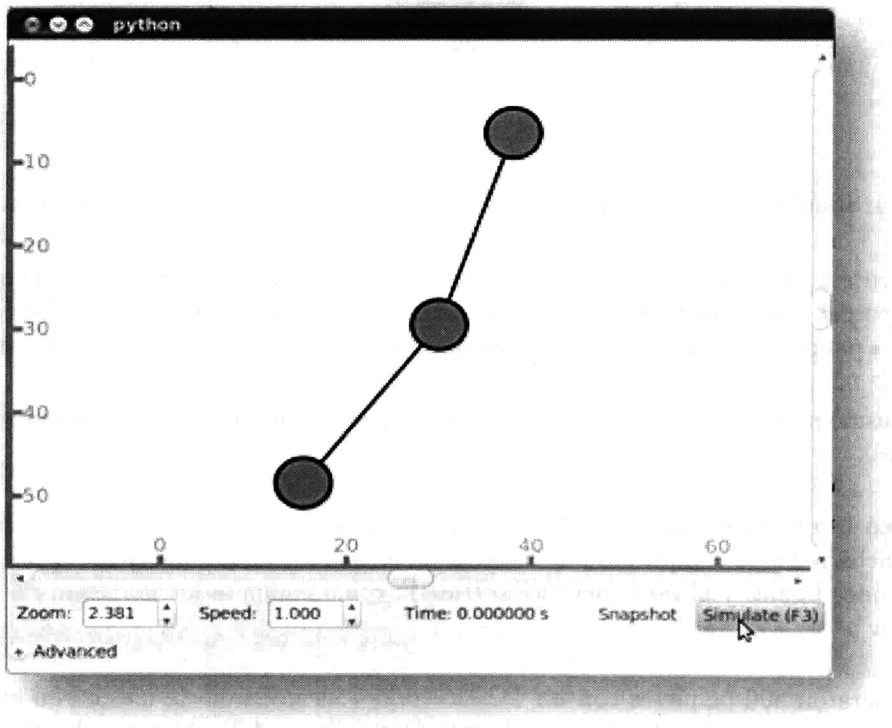


Figura 5.5: Simulación de la red del caso de estudio 2 en NS-3.

Número de Nodos: 2.

- **Tipo de canal:** CSMA.

Numero de conmutadores: 1 (Modulo Openflow)

Número de TCs: 8 (TC0-TC8)

Número de VCs: 3 (VC0-VC3)

En el VC0 (38 % del total de tramas transmitidas) los tiempos variaron desde 0.1144 hasta 3.85981 mseg y obtuvieron una media de 1.763107 mseg. En el VC1 (19 % del total de tramas transmitidas) los tiempos fueron desde 0.89602 hasta 3.05271 mseg con una media de 1.63611 mseg. Por último en el VC2 (43 % del total de tramas transmitidas) el tiempo varió desde 0.89095 hasta 3.3549 mseg con una media de 1.392199 mseg.

5.3. Análisis

Después de realizar las simulaciones de los casos de estudio se observó que en los 5 segundos de simulación se obtuvo una mejora. Se observó en el caso de estudio 1 donde se realizó una red donde se utilizaba un conmutador Ethernet, se obtuvo una media de la latencia de los paquetes de 1.620297 mseg. En dicha simulación no se brindo diferenciación de tráfico por lo que el tráfico de mayor importancia tuvo el mismo trato que la de menor.

En la simulación del caso estudio 2 donde se brindo QoS ("diffserv") por medio de la utilización de la lógica de los conmutadores PCIe en el conmutador Ethernet (capa 2), se observó que los paquetes correspondientes al VC2 (mayor prioridad) fueron beneficiados en el trato ya que obtuvieron una media de tiempo de latencia de 1.392199 mseg (menor que el obtenido en el caso de estudio 1). Por consiguiente se pudo brindar QoS desde Ethernet disminuyendo latencias, números de marcados de paquetes y así mismo relacionando paquetes provenientes de PCIe con tramas Ethernet (Figura 5.7).

Al comparar la media de 1.620297 mseg de la simulación de la red con un conmutador Ethernet con la media de 1.392199 mseg del conmutador Ethernet con las capacidades extendidas de los conmutadores PCIe ("diffserv"), se observo que al implementar nuestro marco de trabajo disminuye la latencia en las tramas transmitidas de forma considerable.

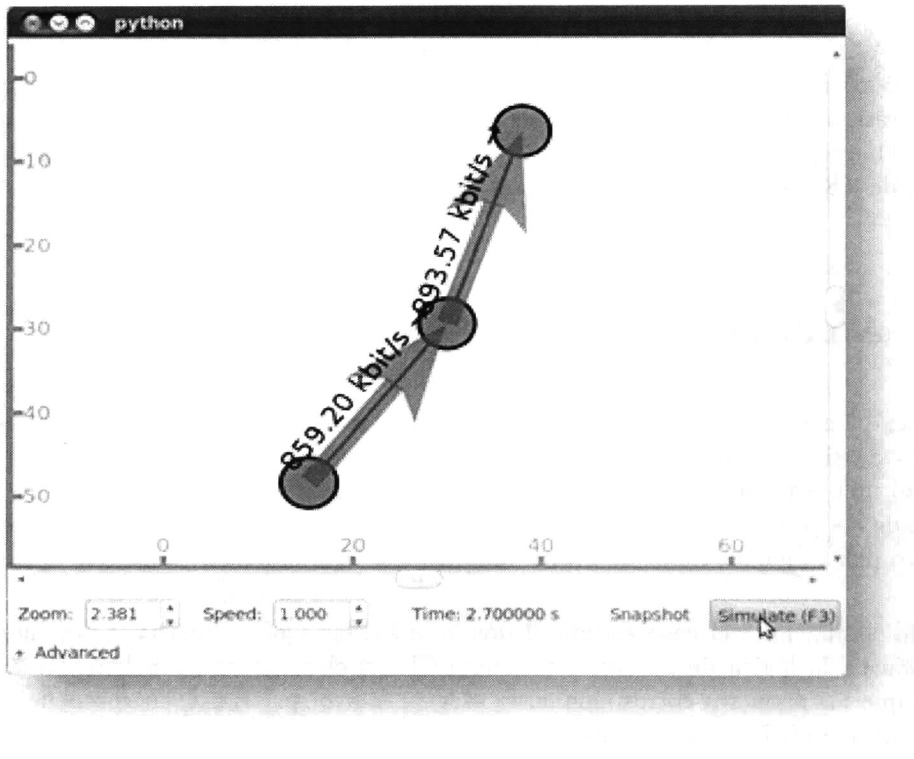


Figura 5.6: Simulación de la transmisión de datos del caso de estudio 2 en NS-3.

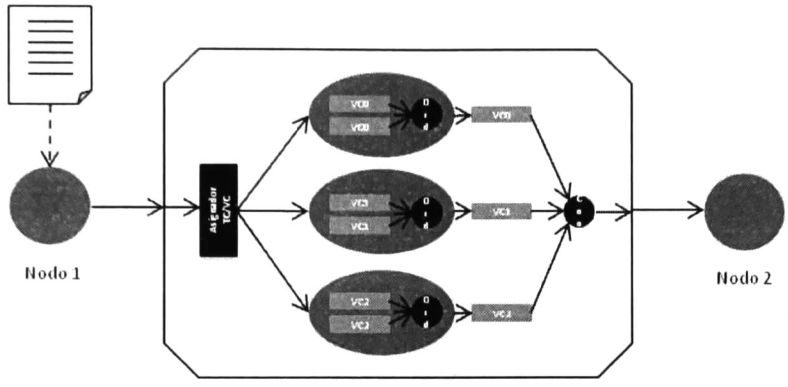


Figura 5.7: Simulación de una red Ethernet, donde se le dan de entrada datos obtenidos de la red PCIe

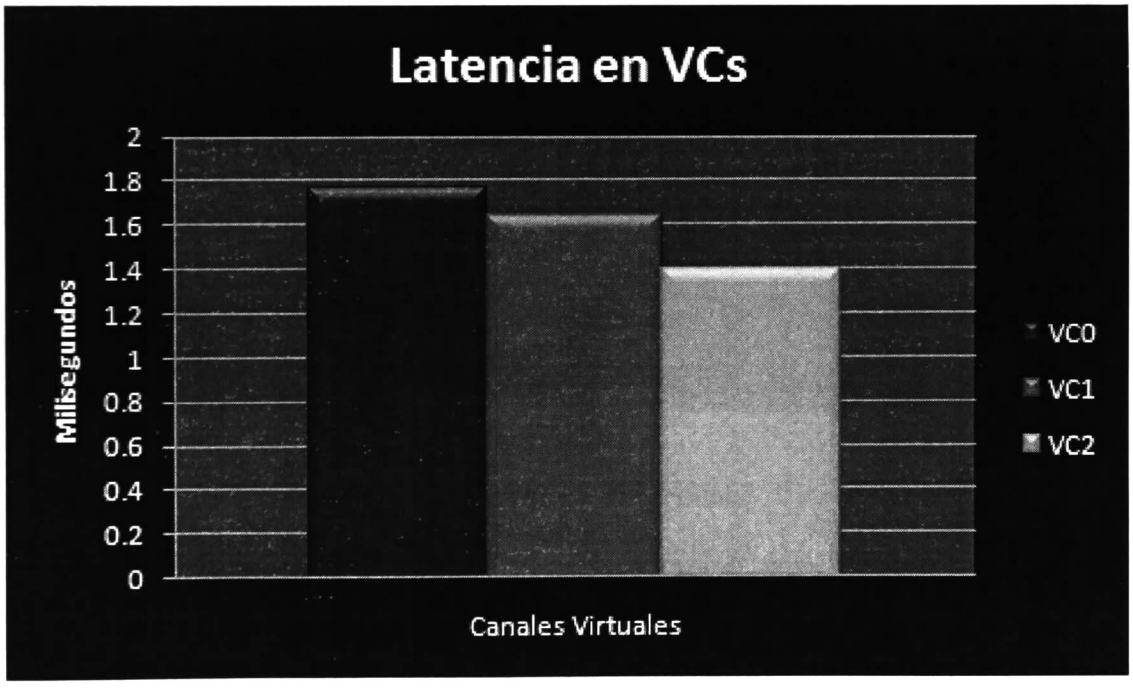


Figura 5.8: Resultados de la media de los tiempos de transmisión de cada VC.

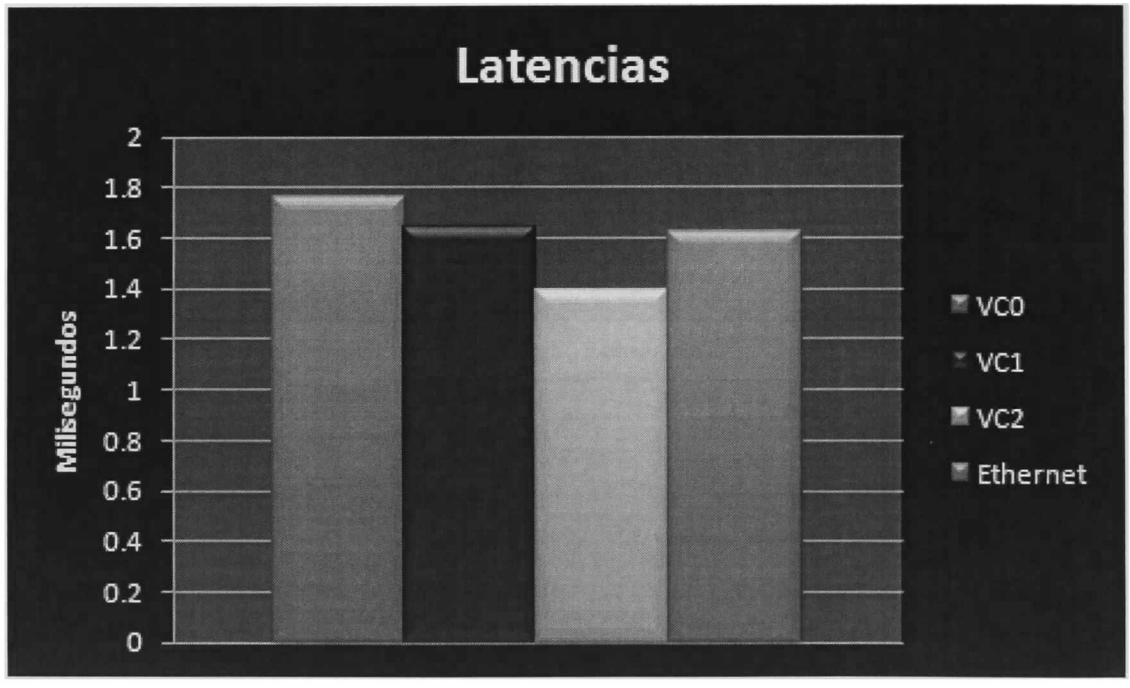


Figura 5.9: Análisis de la latencias entre Ethernet y VCs.

Capítulo 6

Conclusiones y trabajo futuro

En esta sección se muestran las conclusiones a las que llegamos y una lista de trabajos que surgen para complementar esta tesis.

6.1. Conclusiones

La principal aportación de la presente investigación es el marco de trabajo de comunicación fin a fin desde PCIe hasta una red Ethernet. Esto se logró con la ayuda de la extensión de la cabecera de las tramas Ethernet para poder relacionarse con los paquetes de PCIe, así mismo se utilizó la lógica consistente en "diffserv" de los conmutadores aplicados en la tecnología PCIe en los conmutadores Ethernet con el objetivo de brindar QoS desde capa 2 sin tener la necesidad de usar funciones de la capa 3 para poder brindarla.

Utilizando el marco de trabajo con extensión de la cabecera en las tramas Ethernet y la utilización de la lógica ("diffserv") de los conmutadores PCIe en los conmutadores Ethernet, se ha demostrado que, tomando en cuenta las características con las que cuenta la tecnología Ethernet, se logro extender las capacidades de QoS desde el dominio de PCIe hasta el dominio de una red Ethernet.

Se realizaron simulaciones aplicadas de parte de la lógica de QoS en PCIe a Ethernet. Por medio de las pruebas se pudo observar una mejora consistente en los tiempos de latencias de las tramas que se transmitieron utilizando el marco de trabajo propuesto con respecto a las tramas que se transmitieron por medio de un conmutador Ethernet normal. También se probó que se podía disminuir las veces de etiquetado de tramas por medio de la relación de paquetes de PCIe con tramas de Ethernet.

De la revisión bibliográfica, no se encontró un marco de trabajo que brinde QoS para una comunicación fin a fin desde PCIe hasta Ethernet, sin embargo, en esta investigación se demostró que extendiendo las capacidades de QoS desde PCIe hasta Ethernet se puede brindar QoS ("diffserv") obteniendo resultados favorables.

6.2. Trabajo futuro

En esta sección se muestran algunas condiciones para mejorar el presente trabajo de investigación, haciéndolo más flexible y apegado a las condiciones de una red y transmisión reales:

1. Realizar un simulador de PCIe para poder trabajar en capa 2 (enlace de datos)

2. Realizar pruebas utilizando un simulador que cuente con todas las características de PCIe con la finalidad de poder probar completamente el marco de trabajo propuesto.

3. Hacer flexible en número de canales virtuales, en PCIe y en Ethernet así mismo incrementar el número de nodos en las simulaciones a efecto de darle más realismo a las simulaciones.

4. Realizar un marco de trabajo en donde abarque deferentes redes de sistemas embebidos y no este limitado a solo uno.

Bibliografía

[1] A. Forouzan. *Transmisión de datos y redes de comunicaciones.* Mc. Graw Hill, 4 edition, 2007.

[2] J. Bryanti. Pcb design issues. 2000.

[3] A. Wesley. Pci express system architecture. 2003.

[4] W. Dally y A. Towles. *Principles and Practices of Interconnection Network.* Elsevier, 1 edition, 2004.

[5] Ieee 802.3"ieee 802.3 ethernet working group" 2008.

[6] C. Partridge y R. Guerin S. Shenker. Rfc 2212 "specification of guaranteed quality of service" 1997.

[7] J. Wroclawski. Rfc 2211 "specification of the controlled-load network element" 1997.

[8] F. Baker y D. Black S. Blake. Rfc 2474 "definition of the differentiated services field (ds field) in the ipv4 and ipv6 headers" 1998.

[9] Et. All B. Raahemi, G. Chirivolu. Metro ethernet quality of services. *Alcatel Telecommunications Review 4th Quarter 2004*, 2004.

[10] L. Kamoun y L. Chaari H. Mliki. Ethernet congestion manager charasteristics, calibration ana analysis. 2010.

[11] A. Hasib y A. Fapojuwo. A qos negotiation framework for heteregeneous wireless networks. 2007.

[12] V. Krisshnan y D. Mayhew. A localized congestion control mechanism for pci express advanced switching fabrics. 2004.

[13] Et. All C. Bouras, V Kapoulas. Extending qos support from layer 3 to layer 2. 2008.

[14] Et. All F. Obaya, C. Baudoin. Traffic contracts based optimizations for qos support in dvb-rcs satellite systems. 2010.

[15] y J. Duato F. Alfaro, J. Sanchez. Qos infiniband subnetwork. *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEM*, pages 810–823, September 2004.

[16] Intel. The pci express architecture and advanced switching. *Intel*, 2003.

[17] K. Gopalakrishnan y D. Panda A. Koop, W. Huang. Performance analysis and evaluation of pcie 2.0 and quad-data rate infiniband. 2008.

[18] T. Sodring y O. Lysne S. Reinemo, T. Skeie. An overview of qos capabilities in infiniband, advanced switching interconnect, and ethernet. 2006.

[19] IEEE 802.3x. Flow control. 1997.

[20] D. Bergamasco. Ethernet congestion manager. 2007.

[21] T. Divoux y E. Rondeau J. Georges. Strict priority versus weighted fair queueing in switched ethernet networks for time critical applications. 2005.

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N.
UNIDAD GUADALAJARA

"2011, Año del Turismo en México"

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Marco de Trabajo de Calidad de Servicio bajo un enfoque de comunicación fin a fin entre PCIe y Ethernet

del (la) C.

Sagrario Corina QUEVEDO PILLADO

el día 16 de Mayo de 2011.

Dr. Luis Ernesto López Mellado
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. Félix Francisco Ramos Corchado
Investigador CINVESTAV 3A
CINVESTAV Unidad Guadalajara

Dr. Mario Angel Siller González
Pico
Investigador CINVESTAV 2C
CINVESTAV Unidad Guadalajara