

## PC 기반 가상라우터 관련 기술

이종원 (한동대학교)

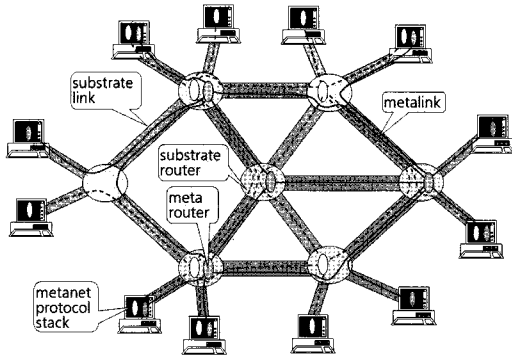
### 1. 서론

TCP/IP 프로토콜을 근간으로 한 인터넷은 1990년대 이후로 급격하게 확산되어 현대 정보화 사회의 기반이 되고 있다. 그 동안 인터넷 트래픽 양이 증가함에 따라 전송 선로 및 라우터 처리용량의 확충도 이루어져 왔으며, 또한 여러 가지 문제가 발생될 때마다 그에 따른 추가적인 보완책이 강구되어, 현재 특별한 문제 없이 많은 사람들이 인터넷을 이용하여 다양한 서비스를 즐기고 있다.

하지만 다른 한편에서는 현재의 인터넷이 성공적으로 사용되고 있지만 여러 가지 면에서 근본적인 제약점을 지니고 있으며, 구조(architecture)적인 면에서 새로운 아이디어를 구현하기 쉽지 않다는 점을 지적하고 있다.<sup>[1]</sup> 이러한 관점에서 현재 인터넷의 현황은 다음과 같이 표현되고 있다. "Internet has become ossified." "Internet is under stress." "Internet is in an impasse." 이에 따라 기존의 인터넷과의 양립성을 고려하지 않은 새로운 인터넷에 대한 연구가 미국을 비롯한 여러 나라에서 진행되고 있다.

이러한 새로운 인터넷(일반적으로 미래 인터넷: Future Internet 이라고 함.)에 대한 연구를 위하여 새로운 구조와 프로토콜 등을 실험할 수 있는 실험망의 필요성이 대두되었다. 이러한 미래 인터넷을 위한 실험망이 지녀야 할 특성으로 여러 연구자가 동시에 서로 다른 구조의 통신망을 구현하여 실험할 수 있는 구조를 지니고 있어야 한다는 점이 지적되고 있다.<sup>[2]</sup> 이러한 구조적인 특성을 지닌 통신망이 현재의 인터넷을 대체하여 궁극적인 미래의 인터넷이 될 수 있을 것인지 단언하기 어렵지만, 인터넷 연구에서 중요한 전기가 마련된 것은 확실하다.

앞에서 언급한 요구 사항을 만족시키기 위하여 제안된 방안이 네트워크 가상화이다. 하나의 네트워크에서 가상화한 물리적 자원을 이용하여 필요 시 기존의 통신망과 논리적으로 분리된 새로운 통신망을 구축함으로써, 동시에 서로 다른 구조의 통신망이 동작할 수 있도록 하는 것이 네트워크 가상화의 동작 특성이다. 이러한 가상화 개념은 최근에 더욱 많은 관심을 끌고 있긴 하지만 이미 컴퓨터 분야에서 오래 전부터 사용되고 있는 기술이며, 네트워크 분야에서도 VLAN, VPN 등에서 사용되고



<그림 1> “Diversified Internet” 구성도.<sup>[2]</sup>

있는 개념이다. 하지만 미래 인터넷을 위한 실험망에서 요구되고 있는 네트워크 가상화는 VLAN이나 VPN에서 사용되고 있는 논리적인 데이터 경로의 분리보다는 더 많은 자원에 대한 가상화가 요구되고 있다.

네트워크 가상화에 대한 개념은 현재 세인트 루이스에 있는 워싱턴대학에서 제안하고 있는 “Diversified Internet”<sup>[2]</sup>이라는 구조를 살펴 보면 이해가 쉬울 것이다. <그림 1>에는 물리적으로는 하나의 네트워크이지만, 논리적으로 두 개의 네트워크로 동작하고 있는 상황을 표현하고 있다. (점선으로 연결된 네트워크와 실선으로 연결된 네트워크) 물리적으로는 라우터(그림에서는 “substrate router”)와 라우터들을 연결하고 있는 링크(그림에서는 “substrate link”)들로 구성된 하나의 네트워크이지만, “substrate router”가 가상화되어 있어서, 필요시 다수의 독립된 기능의 라우터(그림에서는 “meta router”)로 동작할 수 있다. 이러한 “meta router”들을 직접 연결하는 링크는 “metalink”로 표시되어 있으며, 이는 물리적인 링크인 “substrate link”를 논리적으로 구분한 것이

다. 이상적으로는 한 “substrate link”에 포함된 “meta link”를 통하여 전달되는 트래픽 간에 간섭이 없어야 할 것이다.

앞의 예에서 본 바와 같이 미래 인터넷을 위한 실험망에서 요구하는 네트워크 가상화를 위해서 가상화된 라우터가 필수적이다. 대용량의 처리 능력을 위해서는 네트워크 프로세서 등과 같은 특별한 하드웨어의 지원이 필요하겠지만, 일반 PC와 같은 보편적인 하드웨어를 이용하여 가상화 라우터의 구현하는 것에 대한 관심도 증대하고 있다. 이는 경제적인 문제도 있지만, 요즘 향상된 PC의 성능으로 인하여 PC로 구현한 라우터로도 기존의 에지 라우터에서 처리할 수 있는 정도의 트래픽을 처리할 수 있으며, 여러 가지 공개 소프트웨어를 이용하여 많은 노력 없이 가상화된 라우터를 구현할 수 있기 때문이다.

실험실 수준에서 PC를 라우터로 동작하도록 하여 실험에 활용하는 예는 이전부터 적지 않았으며, 최근 읍살라대학에서는 PC(서버급)를 이용한 라우터를 ISP와 연결되는 라우터로 활용하여 학내망을 운영하고 있다. 약 30,000명의 학생들을 수용하고 있는 UpUnet에서 학생들을 위한 코어 라우터는 (하드웨어로는 2개의 AMD Opteron 2220 2.8GHz CPU, 메모리 2 GB, 인텔 기가비트 이더넷 NIC로 구성되어 있으며, 소프트웨어로는 Bitfrost Linux 배포판과 Quagga 라우팅 소프트웨어를 사용하고 있음) 패킷 손실 없이 기가비트 속도로 패킷을 처리하고 있는 것으로 보고되고 있다.<sup>[3]</sup>

본고에서는 PC와 같은 일반적인 하드웨어를 이용하여 가상화된 라우터를 구현하는 기술에 대하여 소개하고자 한다. II장에서는 PC 기반의 가상 라우터를 구현하는 데에 현재 적용되고 있는 기술에 대해서 소개하고, III장에서는 PC 기반 가상라우터를 구현하고 있는 예를 소개한다.

IV장에서는 PC 기반 가상 라우터에서의 포워딩 성능에 대한 간단한 실험 결과를 소개하고, V장에서 추후 전망에 대해서 살펴본다.

## II. PC 기반 가상라우터의 요소 기술

라우터를 구성하기 위해서는 기본적으로 라우팅을 위하여 라우팅 소프트웨어와 패킷 포워딩을 위한 포워딩 엔진이 필요하다. 가상라우터를 만들기 위해서는 하드웨어 자원을 가상화시킬 수 있는 소프트웨어가 추가로 필요하게 된다. 이 장에서는 가상라우터를 구현하기 위하여 사용할 수 있는 공개 소프트웨어들을 살펴본다.

### 1. 가상화 기술

서버에서의 가상화(virtualization)란 컴퓨터의 자원을 추상화시킴으로써 다른 사용자 혹은 프로그램이 하드웨어의 자세한 부분을 직접 다루지 않고도 사용할 수 있게 하는 기술이다. 서버의 가상화는 오늘날의 컴퓨팅 환경에서 대단히 중요한 부분이 되었지만 그 개념은 새로운 것은 아니다. 가상화의 개념은 1960년대 IBM 메인 프레임 컴퓨터에 적용되기 시작하였다. 그 당시의 관심사는 기존의 안정화된 OS와 함께 베타 버전의 새로운 OS를 동시에 한 컴퓨터 시스템에서 동작시킬 때, 베타 버전 OS가 버그에 의해서 작동을 중단하더라도 기존 OS에 의해서 동작되고 있는 프로그램은 영향을 받지 않도록 하기 위한 것이었고, 그런 관점에서 가상화 기술이 도입되었다. 1990년대에 스탠포드대학에서 x86 플랫폼에 대한 가상화에 대한 연구가 시작되었고, 이 연구의 결과로 VMware 사가 창설되었고, 그 회사에 의

해서 x86에 대한 첫 상용화된 가상화 제품이 2001년도에 출시되었다. 하지만 x86 기반의 CPU에서는 가상화를 위한 하드웨어 지원이 없었기 때문에, 특정 명령어를 직접 변환(binary translation)시켜 실행하여야만 하였다. 2005년도 이후로 인텔사와 AMD사에서 가상화를 지원하기 위한 CPU로 인텔에서는 IntelVT를 개발하였으며, AMD에서는 AMD-V를 개발하였다. 최근에 출시된 대부분의 CPU들은 하드웨어에서 가상화를 지원하고 있으며, 가상화를 지원하는 CPU를 사용하는 경우에는 기존의 OS를 수정함이 없이 사용할 수 있는 전가상화(full virtualization)가 가능하게 된다.<sup>14)</sup>

가상라우터의 구현 관점에서 주요한 가상화 구현 방법으로는 전가상화(full-virtualization), 반가상화(para-virtualization), OS 수준의 가상화(OS-level virtualization)가 있다. 가상화 방법에 따라 게스트 시스템 간에 사용 자원에 대한 영향력 분리(isolation)의 정도에 차이가 있게 된다. 일반적으로 분리의 정도를 크게 하려면 그러한 기능 구현을 위한 대가로 성능의 열화가 수반된다. 역으로 분리의 정도를 완화시키면 성능을 높일 수 있다.

#### 가. 전가상화

전가상화는 게스트 시스템의 OS가 수정됨이 없이 실행될 수 있으며, 각 게스트 시스템이 실제 하드웨어와 동일한 형태의 가상화된 장치 상에서 실행되게 된다. 전가상화 시스템에서는 하이퍼바이저가 각 게스트 시스템에게 자신만의 독립적인 하드웨어를 가지고 동작하는 듯한 환경을 제공한다. 가상화를 위한 하이퍼바이저는 그 하드웨어에 대한 접근을 제어하는 기능을 수행한다. 전가상화를 지원하는 대표적인 예로는 VMware 사의 GSX 서버, MS 사의 Hyper-V, ZEN

등이 있다. 하지만 ZEN은 가상화를 지원하는 CPU (Intel-VT, AMD-V) 하에서만 전가상화가 가능하다. 전가상화는 일반적으로 반가상화보다는 성능이 떨어지나, OS를 수정 없이 사용할 수 있다는 장점이 있다. 전가상화의 일반적인 단점으로는 I/O 작업을 많이 요구하는 프로그램에서의 성능 열화가 특히 많은 것으로 알려져 있다.

## 나. 반가상화

반가상화의 경우에는 전가상화와 유사하나, 게스트 시스템의 지원을 좀 더 간단하게 하기 위하여, 게스트 OS에 변경이 요구된다는 특징이 있다. 하지만 이러한 경우 게스트 OS로 소스가 공개되어 있지 않은 Window와 같은 OS를 사용하는 것이 불가능하며, 소스가 공개되어 있다고 하더라도 일반적인 사용자가 자신의 작업에 의해서 원하는 버전의 OS를 게스트 OS로 사용하는 것은 거의 불가능하다. 이러한 반가상화의 대표적인 예로는 UML(User Mode Linux)과 ZEN이 있다.

## 다. OS 수준의 가상화

이 방법은 paene-virtualization 이라고도 불리며, 이 의미는 'almost virtualization'이다. 이러한 가상화에서는 하이퍼바이저를 사용하는 것이 아니라, 일반적인 OS 내에서 게스트 간에 명백한 분리가 이루어지도록 동작하게 된다. 즉 각 게스트는 자신만의 파일 시스템, IP 주소와 시스템 설정이 가능하게 되므로, 마치 별도의 시스템에서 동작하는 듯한 느낌을 받지만, 다른 경우와는 달리 게스트 시스템은 자신의 독립적인 OS를 가지고 있지 않다는 특징이 있다. 따라서 OS 수준의 가상화는 다른 종류의 가상화에 비해서 게스트

간에 분리(isolation)가 약하다는 단점이 있다. 하지만 모든 게스트가 하나의 관리 체계 하에 있다면 관리자가 자원의 분배를 조정할 수 있기 때문에 이러한 단점을 완화시킬 수 있다. 하지만 가상화를 위하여 요구되는 메모리의 양이 적고, 오버헤드가 적기 때문에 빠르다는 장점이 있다. 대표적인 예로는 OpenVZ, Linux- VServer, Solaris Containers, FreeBSD Jails 등이 있다.

현재 자료에 의하면 PC 기반의 가상라우터를 구현하는 데에 있어서 Xen과 VServer가 사용되고 있다.<sup>68</sup> Xen은 공개된 소프트웨어로 최근에 가장 활발한 연구가 진행되고 있으며, 관련 자료 또한 풍부하기 때문에 가상라우터에 적용하는 데에 장점이 있는 것 같다. VServer는 OS 수준의 가상화 기법을 사용하고 있기 때문에 Xen에 비하여 성능이 우수할 가능성이 있으며, 기존의 PlanetLab에서 VServer를 사용해 오고 있기 때문에 이를 기반으로 하여 성능을 향상시키려는 노력이 진행되고 있다.

가상라우터 구현 시 VServer를 선택하려고 할 때 고려하여야 할 점은 VServer가 네트워크 성능을 고려하여 네트워크 부분에 대해서는 완전한 가상화를 하지 않고 있다는 점이다. 즉 모든 게스트 시스템들이 라우팅 테이블과 IP 테이블 등 네트워크 부분을 공유하고 있으며, 단지 자신에게 부여된 IP 주소에만 각 게스트가 구별되어 바인드할 수 있다는 점이다. 따라서 각 게스트에 따라서 다른 네트워크 스택이 필요할 경우에는 각 게스트 시스템의 사용자 레벨에서 이를 처리하여야 한다는 점을 고려하여야 한다. 따라서 OS 수준의 가상화 기법을 이용하면서 네트워크 부분에 대한 가상화가 되어 있는 OpenVZ를 가상라우터 구현에 적용하는 것에 대해 관심이 있는 연구 그룹도 있는 것 같다.

Xen과 VServer/OpenVZ 등은 일반적인 서버 가상화를 목적으로 구현되고 있기 때문에, 어떤 기법을 선택하여 가상라우터를 구현한다 하더라도 라우터의 성능 최적화를 위하여 연구가 진행되어야 할 부분이 많을 것이며, 이에 대한 연구 또한 진행되고 있다.<sup>5,7)</sup>

## 2. 포워딩 관련 소프트웨어

패킷을 포워딩하기 위해서 일반적으로 사용되는 소프트웨어는 리눅스 커널에 포함되어 있는 네트워크 관련 소프트웨어와 Click<sup>8)</sup>이다.

리눅스에서 네트워크 관련 소프트웨어는 커널에서 처리되고 있으며, 라우터로써 동작하기 위한 포워딩 동작은 IP 계층에서 처리된다. 수신된 패킷은 ip\_rcv\_finish() 루틴에 도달하게 되고 그 루틴에서 패킷이 해당 호스트를 목적지로 하는 패킷인지 아닌지가 판명된다. 만약 해당 호스트를 목적지로 하는 패킷이라면 ip\_local\_delivery() 루틴에 넘겨져 상위 계층에서 해당 패킷을 처리하게 하고, 아닐 경우에는 그 패킷을 목적지로 하는 곳으로 전달시켜 주어야 한다. 그러한 포워딩 기능을 수행하려면 해당 머신이 포워딩 기능을 수행하도록 설정되어 있어야 한다. 포워딩을 위해서는 먼저 ip\_route\_input() 루틴에 의해서 FIB (Forwarding Information Base)로부터 목적지를 위한 경로에 대한 정보를 찾는 다음, ip\_forward() 루틴에 넘겨져 포워딩이 되는 과정을 거치게 된다.

앞에서 간단히 살펴본 바와 같이 리눅스에 포함되어 있는 소프트웨어는 라우터를 위한 목적 이라기 보다는 수신된 패킷을 처리하는 일반적인 패킷 처리 과정 중의 한 가지로써 포워딩 기능을 수행하고 있음을 알 수 있다. 따라서 라우터만을 위해서는 최적화 과정을 통해 성능 향상의

가능성이 있음을 예상할 수 있다.<sup>9)</sup>

Click은 라우터, 스위치, NAT 등의 패킷 처리 장치를 구현하는 데에 사용될 수 있는 소프트웨어 툴킷으로 모듈화된 구조로 되어 있다. Click은 C++로 구현되어 있고, 단순한 기능을 지닌 'element'라고 불리는 모듈들로 구성되어 있다. 이미 Click에 포함된 'element'를 이용하여 라우터와 같은 네트워크 장비를 구현할 때에는 다시 컴파일할 필요가 없이, 'element'간의 연결에 대한 부분과 각 'element'의 인자를 스크립트 파일로 작성하여 실행시킴으로써 해당 기능을 구현할 수 있다는 장점이 있다.

Click은 커널 모듈로써 동작시킬 수도 있고, 커널 모드가 아닌 사용자 모드에서 동작시킬 수도 있다. 두 모드에서는 (일부 동작이 한 모드에서만 되는 'element'도 있기는 하지만 대체 가능한 다른 'element'가 존재하기 때문에) 기능 상의 차이는 없고, 단지 성능에서의 차이만이 존재한다. 즉 사용자 모드에서 동작 시에는 커널 메모리에 있는 내용을 사용자 메모리로 복사를 하여 사용해야 하기 때문에 성능의 열화는 피하기 어렵다.

리눅스에 포함된 소프트웨어를 이용하여 구현한 라우터와 Click을 이용하여 구현한 라우터의 포워딩 성능이 비교된 결과가 발표된 적이 있다.<sup>10)</sup> 비교적 단순한 상황에서의 결과이지만 두 가지 모두 기가비트 정도의 포워딩 능력을 지니고 있음을 보여 주고 있으며, 대부분의 특성에서 Click으로 구현한 라우터가 리눅스 자체에 포함된 소프트웨어를 이용한 라우터에 비해서 비슷하거나 우수한 성능을 나타내고 있음을 보인다. 하지만 제한된 환경에서 일부 기능만을 최적화시켜 한 실험이므로, Click으로 구현한 라우터의 성능이 항상 더 우수할 것이라는 의미보다는 Click으로 구현한 라우터도 우수한 성능을 나타

낸다는 정도로 해석하는 것이 안전해 보인다.

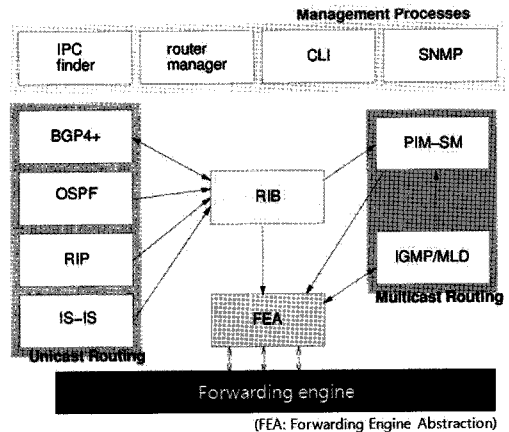
### 3. 라우팅 관련 소프트웨어

이전에 주로 많이 사용한 공개 라우팅 소프트웨어는 Quagga<sup>[11]</sup>였으나, 최근 들어서는 XORP(eXtensible Open Router Platform)<sup>[12]</sup>가 많은 관심을 끌고 있다. Quagga는 이전의 GNU Zebra로부터 파생된 것으로, 다음과 같은 데몬(daemon)으로 구성되어 있다. 기본적인 관리를 하는 zebra, OSPFv2를 구현한 ospfd, RIP v1과 v2를 구현한 ripd, OSPFv3(IPv6)를 구현한 ospf6d, RIPng(IPv6)를 구현한 ripngd, BGPv4+(multicast와 IPv6도 지원)를 구현한 bgpd로 구성되어 있다. Quagga는 다른 프로토콜을 쉽게 구현하게 하기 위한 풍부한 개발용 라이브러리를 지니고 있다는 장점이 있다.

XORP는 네트워크 연구자들을 고려하여 특히 확장성에 중점을 두고 2000년부터 Mark Handley에 의해서 개발되기 시작한 라우팅 소프트웨어이다. XORP의 특징은 <그림 2>에 잘 나타나 있다.

XORP 또한 C++로 구성되어 있으며, <그림 2>에 표시된 RIP, OSPF, BGP4+ 등의 각각의 라우팅 프로토콜과 관리를 위한 기능들이 별도의 프로세스로 실행되는 구조로 되어 있으며, 확장성을 용이하게 하게 위하여 각 모듈 간에 XRL(XORP Resource Locator)를 이용하여 프로세스 간 통신을 하도록 구성되어 있다.

각 라우팅 프로세스는 자신이 얻은 경로에 대한 정보를 RIB에 전달하고, 이 정보들은 다른 라우팅 프로세스를 통하여 재분배(redistribution)되기도 한다. XORP의 또 다른 특징 중의 하나는 FEA(Forwarding Engine Abstraction)을 가지고 있다는 것이다, FEA의 기능은 FIB 정보를 추상화시



<그림 2> XORP의 개념적인 구성도

킴으로써 임의의 포워딩 엔진과의 접속을 용이하게 하기 위함이다. 이와 같은 구조는 각각의 라우팅 프로토콜이(<그림 2>에 나타난 것 혹은 사용자가 새로 구현한 프로토콜) 다른 호스트에서 실행되고 있다고 하더라도 그 결과를 FEA를 통하여 어렵지 않게 포워딩 엔진에 반영시킬 수 있다는 장점을 지니게 되고, 이와 같은 특징은 특히 가상라우터를 구현을 용이하게 하는 특징을 지니고 있다.

Quagga와 XORP 모두 동작의 안정성에 대해서는 문제가 없는 것으로 알려져 있다.

### III. PC 기반 가상라우터 프로젝트에

미래인터넷의 실험망을 위한 플랫폼 구현을 세계 여러 연구 단체에서 진행하고 있다. 알려진 프로젝트 중에서 대표적인 것으로는 다음과 같다. 세인트 루이스의 워싱턴 대학에서 진행하고 있는 SPP(Supercharged PlanetLab Platform)<sup>[13]</sup>가 있으며, 목표는 PlanetLab에 적용 가능한 고성능의

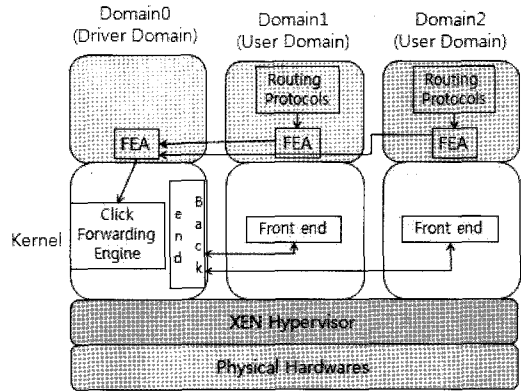
플랫폼을 만드는 것이고, HW는 ATCA 기반의 IXP2850 네트워크 프로세서를 사용하여 구현하고 있다. 현재 관심을 끌고 있는 다른 플랫폼은 스탠포드 대학에서 진행하고 있는 OpenFlow 스위치이다.<sup>[14]</sup> 이는 라우터 제조업자의 입장을 반영하여 플로우 테이블의 내용만을 표준화함으로써 라우터의 내부 구조 공개를 하지 않고 가상라우터를 구현하는 접근 방식으로, 시스코 및 주니퍼를 비롯한 주요 라우터 제조업체의 지지를 받고 있는 상황이다. OpenFlow 스위치는 소프트웨어를 이용하여 구현될 수도 있지만, 현재 스탠포드 대학에서 Xilinx사의 Vertex-II 기반의 NetFPGA라는 하드웨어를 이용하여 구현되고 있다.

PC 기반의 가상화 플랫폼에 대한 연구는 발표 자료에 의하면 영국의 UCL과 Lancaster 대학에서 진행하고 있는 VRouter와 아직 버전이 0.1에 지나지 않고 있지만 미국의 프린스턴 대학과 조지아 공대에서 진행하고 있는 VINI-Trellis가 있다. 본장에서는 이 두 플랫폼에 대해서 간단히 살펴본다.

### 1. VRouter<sup>[15]</sup>

VRouter 플랫폼에서는 기본적으로 가상화를 위하여 Xen을 사용하고, 패킷 포워딩을 위해서는 커널 모드에서 동작하는 Click을 사용한다. 라우팅 소프트웨어로는 XORP와 Quagga를 모두 사용할 수 있는 것으로 기술하고 있으나 XORP를 더 선호하는 것으로 보인다. VRouter의 구성은 포워딩 엔진인 Click을 어디에 설치하는가에 따라 현재 3가지 구성 시나리오로 되어 있으며, 그 중 포워딩 엔진이 Domain0에만 설치되어 있는 구성 방안이 <그림 3>에 나타나 있다.

성능 평가 결과 VRouter는 예지에서 가상라우터로서 사용할 만한 성능을 나타내는 것으로 발



<그림 3> VRouter의 구성 예

표되었다. 하지만 생성된 가상라우터로 이루어진 가상-네트워크는 다른 가상 네트워크와 트래픽 처리에 있어서 영향을 주지 않아야 한다는 점 등에 대한 연구가 필요할 것이다.

### 2. VINI-Trellis<sup>[16]</sup>

VINI-Trellis는 PL-VINI의 성능을 향상시키고자 하는 목적으로 시작되었다. 즉 PL-VINI는 사용자 모드에서 패킷 포워딩을 하기 때문에 만족할 만한 성능을 보이지 못했다. 따라서 커널에서 패킷 포워딩을 구현하여 성능 향상을 꾀하려고 하고 있다. 현재 버전이 0.1에 불가하여 앞으로 정립되어야 할 부분이 많지만, 현재로는 가상화를 위하여 VServer를 사용하고, 네트워크 부분에 대한 가상화를 하지 않은 VServer의 단점을 보완하기 위하여 NetNS를 사용하고 있다. 포워딩 엔진으로는 (명백하게 언급되어 있지는 않지만) 리눅스 자체의 스택을 사용하는 것으로 보인다. VINI-Trellis는 OS 수준의 가상화 기술을 사용하여 PC 기반의 고성능 가상라우터를 구축하려는 시도로 앞으로 결과를 주목할 필요가 있을 것 같다.

## IV. PC 기반 가상라우터의 성능

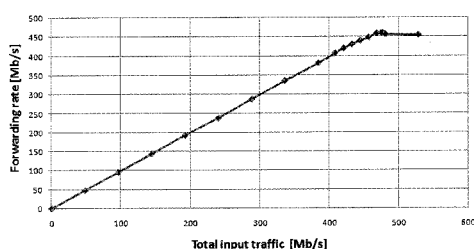
본 장에서는 지면 관계 상 간단하게 PC 기반 가상라우터 플랫폼에서 측정된 결과를 기술한다. Xen 가상화 환경에서 Domain0의 커널에 Click을 설치한 환경에서 기본적인 특성에 대한 측정 결과이다. 사용한 PC는 Dell Power Edge 2900으로 구성은 다음과 같다. CPU는 quad-core Xeon Pro E5405이며 2개가 장착되어 있고, 각각의 동작 클럭 속도는 2GHz이다. 12MB Cache,, 1333MHz FSB의 특성을 지니고 있으며, NIC는 PCIe-4 인터페이스를 지닌 Intel Pro 1000 VT Quad-port이다. 이 NIC에서 사용하는 컨트롤러는 Intel 82575EB이며, 이 NIC를 위하여 사용한 디바이스 드라이버는 폴(poll) 디바이스를 지원하도록 패치된 igb이다.

성능 측정을 Click의 UDPgen과 AverageCount를 사용하여 트래픽 생성과 분석을 하였으며, 모든 실험은 최소 패킷 크기인 64바이트에 대하여 수행하였다. 실험에 의하면 GbE 한 포트에 대해서 Dell 2900의 최대 전송율은 714.288 Mb/s이며, 전송 시에는 디바이스로 PollDevice를 사용하거나 FromDevice를 사용하더라도 성능의 차이가 거의 없었다. GbE 한 포트에 대한 최대 수신율은 PollDevice로 사용할 때는 936.240Mb/s이며, FromDevice일 때는 383.376Mb/s로 2배 이상의 차이가 있음을 알 수 있다. 이는 인터럽트의 오버헤드를 제거한 효과 때문이다. 커널 모드가 아닌 사용자 모드에서는 PollDevice는 사용할 수 없고 단지 FromDevice만 사용 가능하다. 이 경우에 있어서 최대 수신율은 378 kb/s로 사용자 모드에서의 성능은 커널 모드에 비교하여 대단히 낮음을 알 수 있다. 따라서 단순히 기능 구현만을 목적으로 사용할 경우에는 사용자 모드에서 Click을

사용하여도 무방하지만, 성능에 대한 고려가 필요할 경우에는 커널 모드로 동작시켜야 함을 알 수 있다.

Click을 이용하여 IPv4 라우터를 구현하여, 2개의 입력 포트로부터 하나의 출력 포트에 포워딩이 되도록 스택틱 라우팅을 적용하였을 경우, 입력 트래픽 양에 따른 포워딩 양은 그림 4와 같다. 최대 처리량은 약 450Mb/s 정도이다.

실질적인 상황에서는 이보다 최대 처리량이 적어지겠지만, 별다른 최적화 과정을 하지 않았다는 점과 실험에서 최소 패킷을 사용하고 있다는 점을 고려하면 PC 기반 가상라우터의 가능성을 긍정적으로 볼 수 있는 것으로 보인다.



〈그림 4〉 Click IPv4 라우터에서 입력 트래픽 양에 따른 포워딩 트래픽 양

## V. 맺음말

본고에서는 미래인터넷의 실험망 구축 시에 필요한 가상라우터를 PC 기반으로 구현하는 기술에 대해서 살펴보았다. 최근에 출시된 고성능의 PC 하드웨어는 실질적으로 에지 라우터로 동작하기에 충분한 성능을 제공하고 있음이 증명되었고, 또한 가상라우터로 사용할 수 있는 성능을 나타내는 것으로 연구 결과가 발표되고 있다. 하지만 미래인터넷에서 요구되는 조건을 만족시



키기 위해서는 기능적인 면에서와 성능에 대한 지속적인 연구가 요구된다.

최근에 멀티 코어 CPU가 보편화되고 점차 코어의 수도 증가하고 있는 추세에 있으며, NUMA 구조의 CPU도 출시되고 있는 등 하드웨어에 대한 향상이 계속되고 있다. 따라서 이러한 기능을 최대한 활용하여 라우터 혹은 가상라우터의 성능을 향상시킬 수 있을 것으로 보인다. 이러한 PC 기반의 가상라우터는 예산에 대한 제약이 있는 대학에서 다양한 연구를 위한 유연성 있는 플랫폼을 제공해 준다는 면에서 특히 유용할 것으로 생각된다.

#### 참고문헌

- [1] Tom Anderson, Larry Peterson, Scott Shenker and Jonathan Turner, "Overcoming the Internet Impasse through Virtualization," IEEE Computer Magazine, April, 2005.
- [2] Jonathan Turner and Davis E. Taylor, "Diversifying the Internet," IEEE GLOBECOM 2005.
- [3] Robert Olsson, Hans Wassen, and Emil Pedersen, "Open Source Routing in High-Speed Production Use," Linux Kongress 2008, Hamburg Oct. 2008.
- [4] Jeeanna N, Matthews, et. al., "Running Xen: A Hands-on Guide to the Art of Virtualization," Prentice Hall, 2008.
- [5] Norbert Egi, Adam Greenhalgh, Mark Handley, Mickael Hoerd, Laurent Mathy, and Tim Schooley, "Evaluating Xen for Router Virtualization," International Workshop on Performance Modelling and Evaluation in Computer and Telecommunications Networks (PMECT), 2007.
- [6] Stephen Soltesz, Herbert Potzl, Marc E. Fiuczynski, Andy Bavier, and Larry Peterson, "Container-based Operating System Virtualization: A Scalable, High-performance Alternative to Hypervisors," Proceedings of the 2nd EuroSys Conference, Lisbon, Portugal, March 2007.
- [7] Guangdeng Liao, Danhua Guo, Laxmi Bhuyan, and Steve R King, "Software Techniques to Improve Virtualized I/O Performance on Multi-Core Systems," ANCS'08, Nov. 2008.
- [8] Eddie Kohler, et al., "The Click Modular Router," ACM Transactions on Computer Systems 18(3), pp. 263-297, August 2000.
- [9] Raffaele Bolla and Roberto Bruschi, "Linux Software Router: Data Plane Optimization and Performance Evaluation," Journal of Networks, Vol. 2, No. 3, June 2007.
- [10] Andrea Bianco, et al., "Click vs. Linux: Two Efficient Open-Source IP Network Stacks for Software Routers," IEEE Workshop on High Performance Switching and Routing (HPSR 2005), Hong Kong, May 12-14, 2005.
- [11] <http://www.quagga.net>
- [12] Mark Handley, Eddie Kohler, Atanu Ghosh, Orion Hodson, and Pavlin Radoslavov, "Designing Extensible IP Router Software," Proceedings of the 2nd USENIX Symposium on Networked Systems Design and Implementation (NSDI) 2005.
- [13] Jonathan Turner, et al., "Supercharging PlanetLab - a High Performance, Multi-Application, Overlay Network Platform Multi-Application, Overlay Network Platform," Proceedings of ACM SIGCOMM, Aug. 2007.
- [14] <http://OpenFlowSwitch.org>
- [15] <http://nrg.cs.ucl.ac.uk/vrouter/>
- [16] Sapan Bhatia, et al., "Hosting virtual networks on commodity hardware," Georgia Tech

Computer Science Technical Report GT-CS-07-10, January 2008.

### 저자소개



이종원

1982년 서울대학교 전자공학과 학사  
1984년 서울대학교 전자공학과 석사  
1996년 서울대학교 전자공학과 박사  
1984년 ~ 1997년 한국전자통신연구원  
1997년 ~ 현재 한동대학교 전산전자공학부 교수  
주관심분야 : 인터넷 프로토콜, 센서 네트워크, 임베디드 시스템