

미래인터넷의 라우터가상화 기술동향

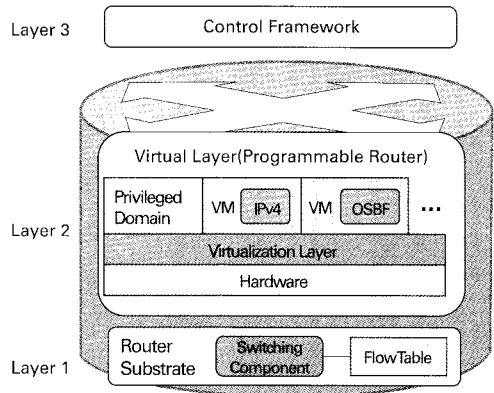
이권용 | 서강대학교 컴퓨터공학과 박사과정
박성용 | 서강대학교 컴퓨터공학과 교수



1. 머리말

미래인터넷이란 현재 인터넷 구조의 근본적인 한계를 해결하기 위해 완전히 새로운 아이디어를 바탕으로 새로운 개념과 개선된 성능을 제공할 수 있도록 인터넷 구조를 새로 설계하는 연구이다. 이는 네트워크 가상화 기술을 기반으로 하여 하나의 거대한 물리적 네트워크망을 동시에 동작하는 여러 개의 가상 네트워크망으로 구성하여 사용자에게 제공하는 것을 목표로 한다. 기존의 네트워크 가상화 기술은 여러 가상 네트워크를 제공하기 위하여 각 네트워크의 트래픽을 분리하는데 초점을 맞추었지만 미래인터넷에서의 네트워크 가상화 기술은 단순히 네트워크의 트래픽을 분리하는 것을 넘어서 각 네트워크 별로 전혀 다른 정책 적용과 동시에 동작할 수 없었던 다양한 프로토콜을 사용할 수 있는 환경을 제공해야 한다. 이러한 미래인터넷에서 가상의 네트워크를 제공하기 위하여 빠질 수 없는 것이 라우터가상화 연구이다. 가상라우터의 기본적인 구조는 [그림 1]과 같이 세 단계로 나눌 수 있다. 계층 1(layer 1)은 라우터에서 패킷의 하드웨어적인 전달(forwarding)을 담당하는 라우터 Substrate, 계층 2(layer 2)는 하드웨어

상위 단계에서 라우터 소프트웨어의 가상화가 이루어질 수 있도록 해주는 계층, 그리고 계층 3(layer 3)은 라우터 노드들을 관리하고 운영할 수 있는 제어 프레임워크(control framework)이다.



[그림 1] 가상라우터 구조

현재 GENT(Global Environment for Network Innovations)[1]의 Spiral 1에서는 라우터가상화의 기본 구조 중 계층 3에 중점을 두고 있지만 다른 분야의 WG(working group)들도 구성되어 제어 프레임워크와 연계되거나 독립적으로 연구들이 진행되고 있다. 그 중 Spiral 1에서 새로이 편성된

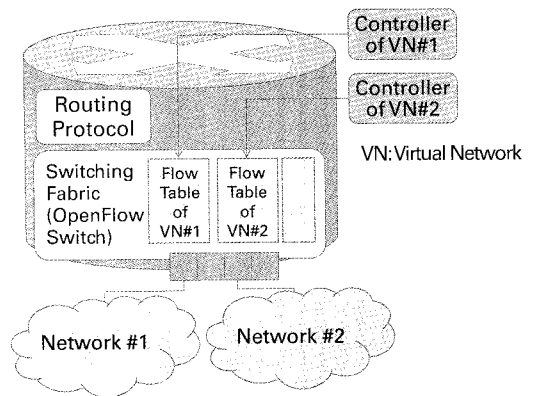
Substrate WG에서는 Substrate를 가상화하는 방안이 연구되고 있으며 Backbone WG에서는 범용 프로세서를 활용한 가상라우터 구축(단계 0)이나 네트워크 프로세서나 FPGA 등을 이용한 라우터 구축(단계 1)을 포함하는 연구들도 진행되고 있다[2]. 본 고에서는 라우터 가상화의 기본 구조 중 계층 1과 계층 2에 해당하는 기술동향을 설명하고자 한다.

2. 하드웨어 기반의 미래인터넷 라우터 구조 연구

일반적으로 라우터나 스위치는 하드웨어 기반의 플로우테이블(flow table)을 이용하여 네트워크 트래픽을 처리하고 있다. 하드웨어 기반의 연구들은 이 플로우테이블을 조작할 수 있는 컨트롤러를 통해 데이터 경로를 설정하여 라우터의 유연성(programmability)을 제공하는 것을 목표로 하고 있다. 하드웨어 기반의 미래인터넷 라우터에서는 네트워크 트래픽의 데이터경로를 하드웨어적으로 설정하고 제어함으로써 각 가상 네트워크에 해당 네트워크의 데이터만이 전송될 수 있도록 가상화 환경을 제공하고 있다. GENI에서는 Substrate WG을 통해 스탠포드 대학의 OpenFlow[3]를 기존 스위치나 라우터에 이식하거나 또는 NetFPGA 등을 활용하여 Substrate를 구축하고자 하는 연구가 Enterprise GENI[4], ProtoGENI[5] 등에서 연구되고 있다. OpenFlow로 사용이 가능한 제품으로는 HP의 ProCurve 5400(5406zl, 5412zl) Ethernet Switch, Cisco의 Catalyst 6500 Series Router, Juniper MX-series, NEC IP8800, NEC WiMax, PC Engines, Quanta LB4G 등이 있다[6,7].

[그림 2]는 Openflow 기반의 라우터 구조를 보여준다. OpenFlow 기반의 라우터는 라우터의 Switching Fabric에 위치한 OpenFlow 스위치의 프로그램 가능

한(programmable) 플로우테이블을 OpenFlow 컨트롤러를 통해 변경할 수 있고 플로우테이블에 의해 결정된 데이터 경로에 따라 가상 네트워크들의 트래픽이 물리적으로 분리되어 가상의 네트워크들을 운영할 수 있게 된다. OpenFlow와 같은 하드웨어 기반의 라우터는 기존의 라우터와 비교하여 성능 저하가 거의 없다는 장점이 있지만 하드웨어적으로 모두 구현이 되어야 하기 때문에 유연성이 떨어져 네트워크마다 서로 다른 다양한 정책을 반영하기가 어렵고 네트워크마다 서로 다를 수 있는 프로토콜들을 동시에 운영하기 어렵다는 문제점이 있다.

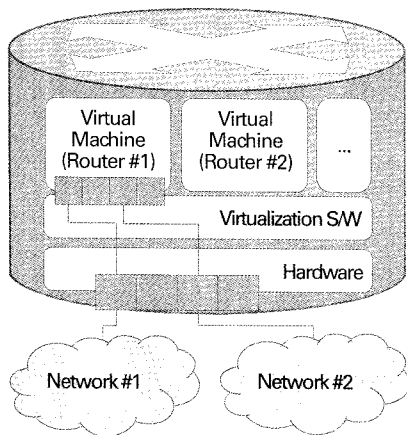


[그림 2] OpenFlow 기반의 라우터 구조

3. 소프트웨어 기반의 미래인터넷 라우터 구조 연구

소프트웨어 기반의 라우터가상화는 [그림 3]에서 보여주고 있는 것처럼 하드웨어 상위 단계에서 가상화 소프트웨어를 통하여 라우터 소프트웨어의 가상화를 제공한다. 가상화 소프트웨어는 VMM(Virtual Machine Monitor)을 통하여 각자 하나의 온전한 기계처럼 동작할 수 있는 다수의 VM(virtual machine)들을 생성하고 관리할 수 있는 소

프트웨어이다. 현재 많은 가상화 소프트웨어들이 존재하지만 오픈 소스이면서 라우터가상화에서 많이 언급되는 소프트웨어로 Linux-VServer[8] 또는 Xen[9]이 있다. GENI에서는 Substrate WG을 통해 PlanetLab[10]의 노드에서 이미 활용되고 있는 Linux-VServer를 이용하여 미래인터넷의 라우터를 구성하고자 하는 연구들이 많이 진행되고 있고 Dome[11]이나 ViSE[12]와 같이 Xen을 이용한 연구들도 있다. 또한 하드웨어 기반의 연구에서 언급한 ProtoGENI에서도 Xen이나 VMware[13]와 같은 가상화 소프트웨어의 이용을 고려하고 있다.

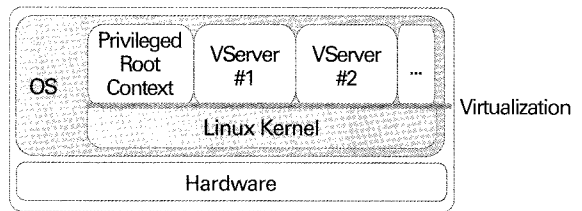


[그림 3] 소프트웨어 기반의 미래인터넷 라우터 구조

3.1 Linux-VServer와 Xen

PlanetLab의 노드에서 사용되는 Linux-VServer는 운영체제 수준의 가상화로 컨테이너(container) 기반의 가상화라고도 하며, [그림 4]에서 보여주고 있는 것처럼 리눅스 커널을 공유하여 여러 사용자에게 서로 분리된 사용자 공간을 VServer라고도 불리는 VM으로 제공한다. 각 VM은 커널을 제외한 작은 공간만을 사용하기 때문에 다수의 VM을 탑재할 수 있다는 장점이 있다. 하지만 커널을 공유하기 때문에 탑재된 VM들이 많은

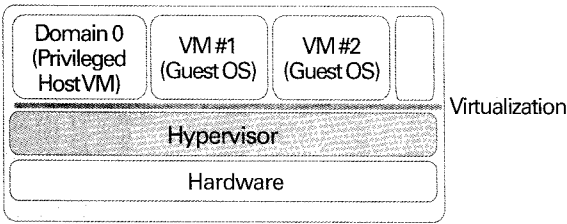
부분에서 하드웨어 자원들의 완벽한 분리가 이루어지지 않는 단점을 갖고 있다. 즉 Linux-VServer를 이용한 라우터에서는 가상 라우터들 간에 커널이 공유되므로 보안 문제가 발생하거나 다른 VM들로부터 간섭을 받게 되는 문제가 존재하여 가상 라우터 간에 완벽한 분리를 요구하는 경우 부적합한 부분이 있다. 또한 모든 VM들은 반드시 Linux를 사용하여야 하는 단점이 있다.



[그림 4] Linux-VServer 구조

Xen은 반가상화(para-virtualization) 기법을 활용한 하이퍼바이저(hypervisor) 기반의 가상화라고도 하며 [그림 5]와 같이 하드웨어 바로 위에서 가상화 계층인 하이퍼바이저의 중계를 통해 VM들에게 완벽하게 분리된 가상의 하드웨어를 제공하여 가상화를 이루어준다. 따라서 각 VM들은 서로에게 영향을 미칠 수 없고 완벽하게 분리된 가상화 환경에서 자신만의 서비스를 제공할 수 있다. 온전히 분리된 가상화 환경을 제공하기 때문에 서로 다른 운영체제를 VM에서 사용할 수 있고 라우터가상화에서 서로 다른 정책이나 다양한 네트워크 프로토콜을 제공할 수 있다. 하지만 VM의 대부분 동작이 하이퍼바이저의 중계를 통해 이루어지므로 병목현상에 따른 성능저하가 발생하게 된다.

Linux-VServer와 같은 컨테이너 기반의 가상화와 Xen과 같은 하이퍼바이저 기반의 가상화를 VM 사이에서 일어나는 영향과 성능저하에 대하여 비교한 결과[14]를 <표 1>에 보여준다. 이는 하나의 잘못된



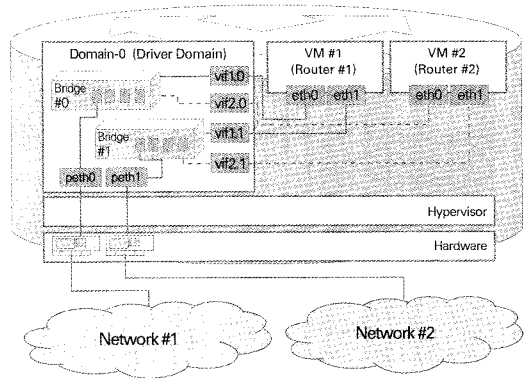
[그림 5] Xen 구조

VM(misbehave VM)으로 인하여 다른 VM들이 받는 영향을 성능 감소의 비율로 나타낸 것이다. 이 결과에 따르면 하이퍼바이저 기반의 가상화는 비교적 훌륭한 VM 간 자원의 분리를 제공하지만 컨테이너 기반의 가상화는 VM 간 분리가 잘 이루어지지 않고 정상적으로 동작하던 VM(well-behave VM)들까지 잘못된 VM으로 인한 충격을 부담하게 된다는 것을 알 수 있다.

3.2 Xen 기반의 라우터 구조

[그림 6]은 가상화 소프트웨어로 Xen을 사용하여 라우터를 구성한 것이다. Xen에 의하여 제공되는 VM들은 각각의 가상환경을 제공받기 때문에 완벽하게 분리된 가상라우터를 구성할 수 있어 가상라우터들 간의 영향은 없다고 볼 수 있다. 하지만 Xen의 네트워킹 기능은 기본적으로 Driver Domain인 Domain 0를 거쳐 이

루어지기 때문에 Domain 0에서 일어나는 패킷 데이터의 복사에 의해 추가적인 CPU 오버헤드나 I/O 성능으로 인한 포화 같은 문제가 발생한다.



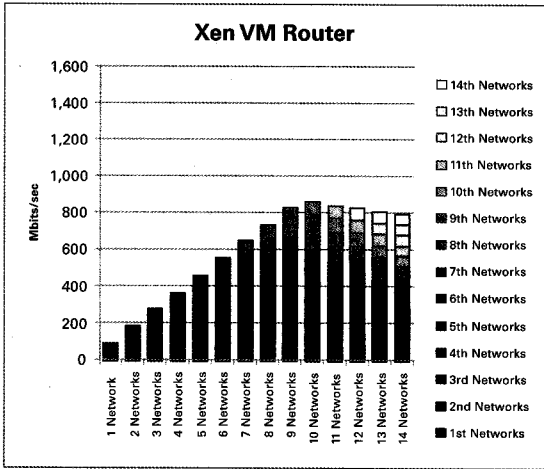
[그림 6] Xen 기반의 라우터 구조

[그림 7]은 Xen의 한 VM이 하나의 100M 네트워크를 구성하는 가상라우터로 연결하여 패킷 포워딩(forwarding) 성능을 측정한 결과이다. 그래프에 의하면 약 10개의 네트워크, 즉 가상라우터 약 10개까지는 운영하는 데 있어서 성능의 저하가 없지만 그 이후 가상라우터가 증가할수록 각 가상라우터의 패킷 포워딩 성능과 라우터 노드 전체의 패킷 포워딩 성능이 감소하는 것을 볼 수 있다.

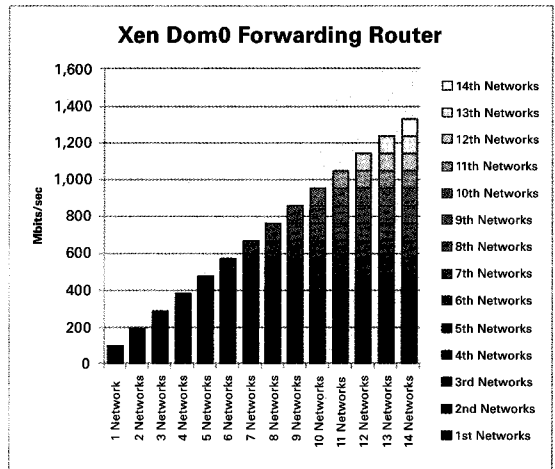
<표 1> 성능 감소 비율 (%)

Test Suite / VMM Type	Hypervisor-based		Container-based	
	Well-behave	Misbehave	Well-behave	Misbehave
Memory test	0,03	DNR	DNR	DNR
Fork test	0,04	DNR	90,03	87,80
CPU test	0	0,03	0	0
Disk Intensive test	1,37	11,53	1,48	1,23
Network Server Transmits test	0,01	0,33	4,00	3,53
Network Server Receives test	0,03	0,10	1,24	1,67

DNR: Does Not Response



[그림 7] Xen 가상라우터의 패킷 포워딩 성능



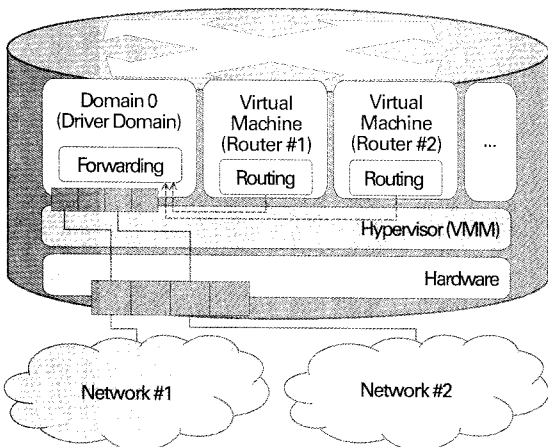
[그림 9] Xen Domain 0 포워딩 성능

Domain 0의 중개에 의한 패킷 포워딩 성능저하를 보완하기 위하여 [그림 8]과 같이 Domain 0에서 모든 포워딩을 수행하도록 구성할 수 있다. 이를 위해서는 Domain 0에서 각 VM에서 운영될 가상라우터들로부터 라우팅 정보를 모을 수 있어야 하고 모아진 정보를 바탕으로 Domain 0가 패킷 포워딩을 하도록 수정되어야 한다. 그 결과 [그림 9]에서 볼 수 있듯이 Xen VM 상의 가상라우터 포워딩 성능 저하를 발생시키는 오버헤드

를 최소화할 수 있다. 앞서 [그림 7]에서 VM의 가상라우터가 10개 이상의 네트워크를 지원하는 경우에 성능의 저하를 겪는 것과는 다르게 각 네트워크의 성능 저하가 없고 Xen 전체의 포워딩 성능 저하도 볼 수 없다.

4. 맺음말

미래인터넷의 라우터가상화 연구는 하드웨어 기반의 가상화와 소프트웨어 기반의 가상화로 나뉘어 진행되어 왔다. 하지만 하드웨어 기반의 연구는 기존의 라우터와 비교하여 성능의 저하가 거의 없지만 하드웨어적으로 모두 구현이 되어야 하기 때문에 유연성이 떨어져 네트워크마다 서로 다른 정책의 반영이 어렵고 네트워크마다 서로 다른 프로토콜을 운영하기가 어렵다는 문제 등이 존재한다. Xen과 같은 가상화 소프트웨어 기반의 연구는 가상화로 인한 오버헤드가 존재하여 포워딩 성능 저하가 발생하지만 완벽하게 분리된 VM 환경을 제공받아 서로 다른 정책을 쉽게 적용할 수 있고 다양한 네트워크 프로토콜을 동시에 지원할 수 있다. 그러나 Xen VM의 가상라우터는 Driver Domain



[그림 8] Xen Domain0 포워딩 라우터 구조

에서 발생하는 병목현상에 의하여 성능저하가 발생할 수 있기 때문에 라우팅은 Xen의 각 VM에서 수행하고 포워딩은 Driver Domain인 Domain 0에서 담당하도록 구성함으로써 병목현상에 의한 성능저하를 줄일 수 있다. 또한 현재 독립적으로 진행되고 있는 하드웨어 기반과 소프트웨어 기반을 함께 구성하여 성능을 위한 하드웨어적인 지원과 소프트웨어 기반의 유연성을 동시에 만족시킬 수 있는 새로운 하이브리드 형태의 미 래인터넷 라우터 구조들이 연구되고 있다.

[참고문헌]

- [1] GENI, <http://www.geni.net/>
- [2] GENI Substrate WG, <http://groups.geni.net/geni/wiki/GeniSubstrate>
- [3] "OpenFlow: Enabling Innovation in College Networks", <http://www.openflowswitch.org/documents/openflow-wp-latest.pdf>
- [4] Enterprise GENI, <http://groups.geni.net/geni/wiki/EnterpriseGeni>
- [5] ProtoGENI, <http://groups.geni.net/geni/wiki/ProtoGENI>
- [6] Jad Naous, David Erickson, G. Adam Convinon, Guido Appenzeller, Nick McKeown, "Implementing an OpenFlow Switch on the NetFPGA platform", ANCS' 08, Nov. 2008.
- [7] http://tiny-tera.stanford.edu/~nickm/talks/infocom_brazil_2009_v1-1.pdf
- [8] Linux-VServer, http://linux-vserver.org/Welcome_to_Linux-VServer.org
- [9] Xen, <http://www.xen.org/>
- [10] PlanetLab, <http://www.planet-lab.org/>
- [11] DOME, <http://groups.geni.net/geni/wiki/DOME>
- [12] ViSE, <http://groups.geni.net/geni/wiki/ViSE>
- [13] VMware, <http://www.vmware.com>
- [14] Todd Deshane, Demetrios Dimatos, Gary Hamilton, Madhujith Hapuarachchi, Wenjiun Hu, Michael McCabe, and Jeanna Neefe Matthews, "Performance Isolation of a Misbehaving Virtual Machine with Xen, VMware and Solaris Containers", USENIX 2006, Feb. 2006. **TTA**