

## 목차

제목: 네트워크 가상화에 대한 고찰 .....	1
1. 서론 .....	1
2. 하드웨어에 기반한 네트워크 가상화.....	1
3. 소프트웨어에 기반한 네트워크 가상화.....	5
4. 앞으로의 방향.....	5

편집: 문수복 (KAIST), 박성용 (연세대)

토론: 김대영 (충남대), 문수복 (KAIST), 박성용 (연세대), 변성혁 (ETRI), 이순석 (ETRI), 신명기 (ETRI), 정일영 (외국어대), -- 가나다순, 이하 성만으로 표기.

## 제목: 네트워크 가상화에 대한 고찰

### 1. 서론

지난 30년간 인터넷은 1980년 “제 3의 물결”에서 앨빈 토플러가 정의한 정보화 사회를 실질적으로 구현하는데 핵심적인 요소의 역할을 해왔다. 기존의 데이터 통신망에 비해서 개방형이고 상위 계층과 하위 계층이 독립적으로 개발될 수 있는 특성으로 인해 비약적인 속도로 확산되어 이제는 전력망, 도로망 등과 같이 없어서는 안되는 사회 간접자본의 일부가 되었다. 인터넷의 비약적인 발전으로 기존의 다양한 매체들이 개별 통신망을 통해 전달되던 방식을 벗어나 하나의 통신망에서 다양한 매체를 전송하는 서비스 집약화의 현상이 급격하게 진행되어가고 있다. 지금까지의 인터넷은 이렇게 고도화된 통신망으로 발전해왔는데 앞으로는 통신망을 어떻게 이용하는가에 제 2의 도약의 관건이 있다. 지금까지 많은 투자를 통해 통신망의 고도화가 이뤄졌다고 한다면, 다음 단계로의 paradigm shift는 단순히 통신망의 스피드를 높이는 것이 아니라, 통신망을 최대한 이용해서 다양한 응용프로그램을 만들어 실생활에 더욱 큰 변화를 이루는 것을 목표로 한다. 즉 단순히 통신망의 Bandwidth를 사용한다는 측면으로 부족하다. 개발자와 연구원들이 통신망의 구조를 깊이 이해하고, 현재의 통신망 구조를 개선하기 위해선, 직접적인 실험과 연구가 필수 불가결하다. 90년대와 같이 ATM (Asynchronous Transfer Mode) network 등과 같이 다양한 전송기술을 기반으로 한 네트워크와 경쟁하던 시대에서 이제는 유일무이한 통신망으로써의 인터넷이 석화(ossification)되지 않으려면 신개념의 시도들이 필요한 때가 온 셈이다.

제2의 paradigm shift 인 통신망 이용 기술의 발전을 위해선, 연구자가 통신망을 들여다 보고, 이를 운용도 해보며, 또한 여러가지 실험을 할 수 있는 환경이 절대적으로 중요하다 할 수 있다. 그러나, 현실적으로 연구자가 기업의 실제 통신망을 운용하는 것은 현실적으로 불가능하고, 차선책으로 실험 연구망을 이용한다 하더라도, 상업용 라우터로 구성된 연구망 라우터를 액세스하거나 자신이 원하는 프로그램을 설치하는 것은 불가능하다. 이를 극복하기 위해 학계에서 의논되고 있는 것이 하드웨어를 통한 네트워크 가상화, 또는 소프트웨어를 통한 네트워크 가상화 등이다. 이런 네트워크 가상화는, 일반 연구자라도 자신만의 통신망을 구성해 새로운 알고리즘과 기능을 구현할 수 있게 하면서도, 다른 사용자들에게는 전혀 영향을 미치지 않고 자유롭게 사용할 수 있는 중요한 연구환경이라 할 수 있다.

이를 위해 우리가 뽑은 필수 요소는 다음과 같다. 실험 및 연구를 위한 가상 네트워크망, 실험 및 개발을 위한 programmable 라우터, 모바일과 센서네트워크를 비롯한 다양한 신규 기술을 유연하게 광대역망에 접속시키는 기술 등이다. 이러한 주제에 대해서 지난 여름 미래 인터넷 포럼 (Future Internet Forum)에서 국내 연구 방향에 대해 활발한 토론이 있었다. 편집인 2명은 ACM SIGCOMM 학회와 PRESTO 워크샵에서 참석하고 있으면서 학회에서 발표된 내용을 요약보고 하면서 토론에 참가하였다. 본 기고문에서는 포럼에서의 토론 내용을 정리하였다. 이하 인용된 문구들은 포럼의 메일링 리스트에서 발췌해서 문맥에 맞게 다듬어 정리한 내용이다.

### 2. 하드웨어에 기반한 네트워크 가상화

#### 2.1. NetFPGA

(신) 최근에 크게 공격적인 데모를 하고 있는 OpenFlow 기술과 관련하여 NetFPGA/OpenFlow set은 당장 Ethernet과 같은 패킷 기반의 diverse한 experiment를 대학등에서 (low cost로) 실험하는데 아주 좋은 short-term solution이다. 예를들어, Type0 (추후에 정의될 Type 1) OpenFlow Switch를 이용하면, 쉽게 패킷 header 들을 rewrite 할수 있어, Indirection architecture (e.g., I3, ID/Loc split), Xcast, IPvX (중국에서 제안한 IPv9 이 실험하기 좋은예일것 같음) 등 IP의 새로운 버전이나 새로운 라우팅, 어드레싱 프로토콜 등을 실험하기에는 아주 용이할 것이다.

(박) 실제로 NetFPGA를 적극적으로 활용하고 있는 Stanford 대학에선 신진 연구 인력과 아이디어 창출이라는 측면에서 상당히 성과를 올리고 있다. Stanford 대학에선 NetFPGA를 이용한, 한 두 학기의 강의 실험을 통해서, 학생들의 이해도를 높이는데 성공하였다. 아래는 지난 8월 4일부터 8일 까지 진행된 NetFPGA Summer Camp 에서 학생들이 완성한 프로젝트이다.

- 802.1q VLANs (top project award)
- Hardware-Accelerated Mathematics Library for NetFPGA (top project award)
- MACinMAC
- Heavy Hitter Identification using Multistage filters
- Layer 2 Load Balancing
- TCP Traffic Analysis for Passive End-to-End Bandwidth Measurement
- Assessment of Prototyping an AFDX Policy Switch Leveraging NetFPGA, Ethane, and OpenFlow Switch

위의 예에서 알 수 있듯이 기존의 프로토콜 (802.1q, L2 load balancing) 과 새로운 규격 (openflow 및 ethane)을 모두 실험 할 수 있다는 점이 매력적이고, 또한 Stanford 학생들의 실력 향상에 이런 프로젝트들이 많은 도움을 주리라는 것을 예상해 볼 수 있다.

(신) 하지만, 이러한 플랫폼은 패킷 스위칭 (플로우) 방식에 제한적이고, NetFPGA는 4port of Gigabit Ethernet에 한정되어 있어, 궁극적으로 GENI와 같이 모든 clean-slate 기반의 실험을 지원해야 하고 전 layer (layer0~app) 에 걸쳐, "Programmable + Virtualizable" 해야 한다는 long-term solution까지 가기 위해서는 많은 challenge를 해결해야 한다.

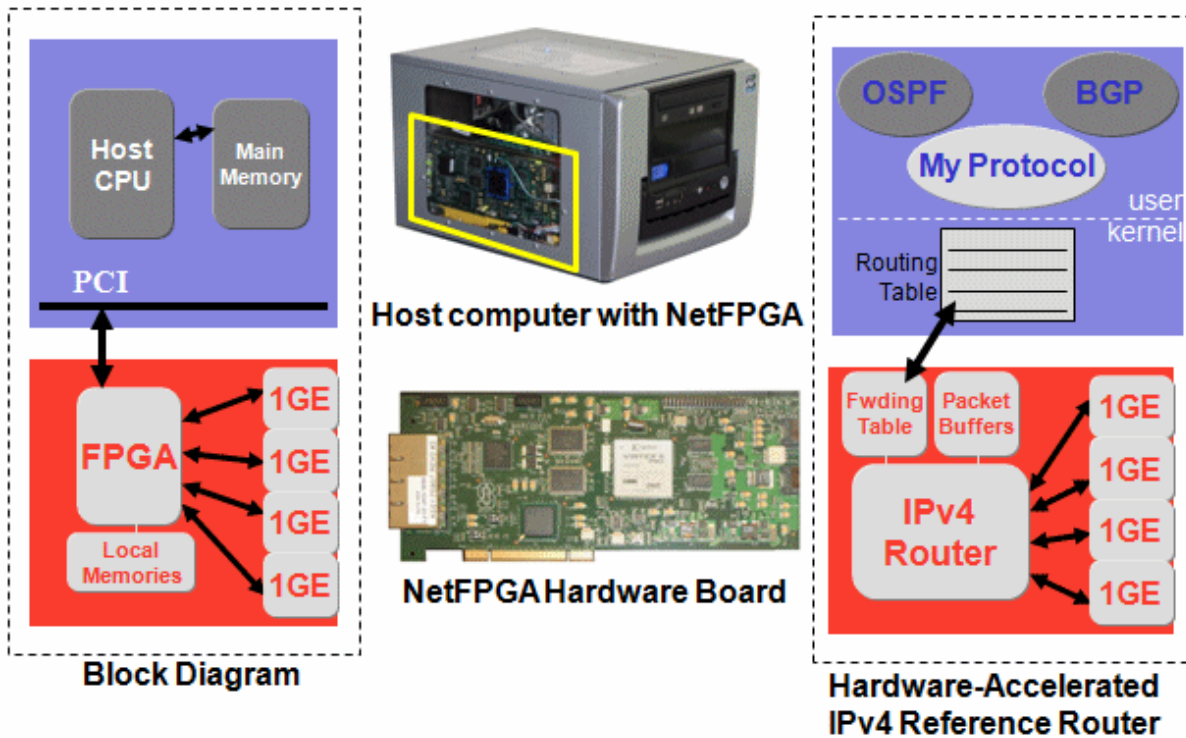
물론 당장 top-down 방식의 구축이 어려울때, 당장 가용한 bottom-up 방식의 이러한 플랫폼을 이용하고 실험하고, 궁극적인 플랫폼 개발을 위해 선행적으로 연구하는 것은 좋은 접근방식이지만, 나중에 실제 GENI Spec이 나오고 하면, long-term 적인 측면에서 Resource Federation하는데 있어, 호환성이 문제될 수도 있다.

또한, John Lockwood 교수도 인정하듯이 현재의 NetFPGA가 표준으로 정해진다는 것은 무리이다. NetFPGA는 미래의 GENI Spec과 기타 연구의 validity를 입증하는 Tool로서 도입하고, 향후 NetFPGA가 얼마나 발전하는 가에 따라 입장을 정하는 것도 좋을 듯 하다.

10월말에 있을 GEC3 회의에서 현재 진행되고 있는 29개의 GENI 관련 prototyping에 대해 발표될 예정인데, 실제 미국 대학등에서 구현이 많은 부분 진행되었더라도 (GEC3 마지막날 OpenFlow 데모가 있을 예정임), WG 차원의 Spec 작업은 아주 더디게 가는 이유도 같은 맥락이라고 생각합니다.

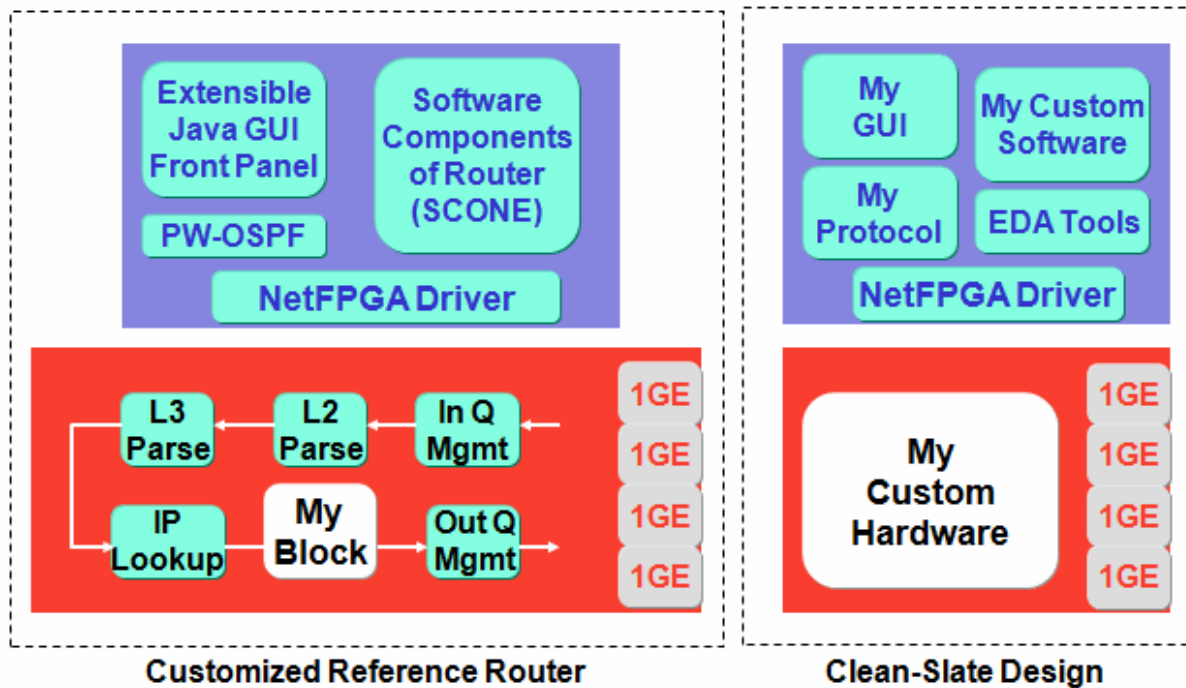
## 2.2 Architecture of NetFPGA

(박) NetFPGA는 본질적으로 4-port Ethernet 카드이다. Ethernet포맷이라는 널리 알려진 표준을 따르기 때문에 처음 접하는 사람이라도 ethernet을 base로 해서 연구하면 되겠다는 실마리를 준다. NetFPGA가 다른 점은 아래 그림의 왼쪽에 나와 있듯이 소프트웨어와 하드웨어의 분리에 있다. 연구자는 소프트웨어에서 유저 Level 및 Kernel level의 프로그램을 통해 자유롭게 새 프로토콜과 실험망을 꾸미고 Kernel level의 Routing Table을 NetFPGA에 있는 Fwding Table 로 전송한다면, NetFPGA는 충실하게 Forwarding Table의 내용에 따라 패킷을 전송한다. 따라서 연구자는 low level의 hardware에 신경쓰지 않고 새로운 software와 알고리즘 개발에 집중할 수 있다.



[출처: [http://netfpga.org/netfpgawiki/images/9/90/System\\_Diagram\\_1.gif](http://netfpga.org/netfpgawiki/images/9/90/System_Diagram_1.gif)]

또한, NetFPGA의 하드웨어 부분은 확장이 가능하게 되어 있어, 새로운 모듈을 만들어 구성하면, 전혀 새로운 구조의 라우터도 구성할 수 있다. 아래의 그림에서 완전 customize 된 구조와 Clean Slate 디자인 예를 보였는데, 이런 구성이 가능하다는 점이 NetFPGA의 장점이다.



[출처:

[http://netfpga.org/netfpgawiki/images/4/41/System\\_Diagram\\_2.gif](http://netfpga.org/netfpgawiki/images/4/41/System_Diagram_2.gif)]

기존의 연구에서는 소프트웨어 라우터를 사용함으로써, 결과에 변수가 많고, 또 결과의 어떤 부분이 SW 때문인지, 어떤 부분이 HW 때문인지 객관적으로 분리/분석하는 것이 힘들는데 NetFPGA를 사용하면 하드웨어 부분의 요소가 measure 가능하기 때문에 상대적으로 SW 적인 요소의 영향을 분석할 수 있지 않을까 기대해본다.

### 2.3 NetFPGA의 확장성

(박) NetFPGA는 하드웨어적인 확장성과 소프트웨어적인 확장성을 어느 정도 만족시켜 준다. 첫번째로, 하드웨어적인 확장성에 대해선, Verilog를 안다면, 원하는 기능을 추가할 수 있다. 2.2장에서 언급되듯, verilog 모듈을 만든다면, non-IP 프로토콜도 지원하게 만들 수 있다. 만일 그렇다면, USN이나 기타 다른 프로토콜을 지원하는 기능도 verilog를 손봐 만들 수 있다. 물론 Verilog에 익숙하다는 전제 하에서 이다.

또, 두번째로 소프트웨어적인 확장성에 대해 언급하자면, NetFPGA의 하드웨어적인 개념이 워낙 간단하기 때문에, 아무리 혁신적인 소프트웨어를 만들어도, 하드웨어 dependency를 고려하지 않아도 되는 점이 확장성에 긍정적인 요소가 된다. 소프트웨어와 하드웨어의 분리(separation)가 잘 이루어진 탓이다.

또, 그 밖의 확장 요인으론 Openflow Switch를 지원할 수 있다는 점이다. 여러 장의 NetFPGA 카드를 PCI slot에 설치하고, 한 장은 OpenFlow용, 다른 한 장은 일반 라우팅 프로토콜 용, 이런 식으로 할당할 수 있다.

이상 확장성에 대해 논의를 마치고, 여담을 전하자면, Cisco Catalysis 6500, HP Procurve 시리즈, NEC의 라우터가 Openflow를 지원하는 연구를 하고 있는데 이번 전시회에도 Cisco 6500에 OpenFlow 기능을 넣은 제품을 가지고 와서 데모를 했다고 한다. Non-Disclosure Agreement (NDA) 때문에 사용자 매뉴얼을 보여주거나, Openflow용 IOS를 줄 수는 없다고 한다. 다만 OpenFlow IOS를 사용하면 OpenFlow에서 정의하고 있는 라우터 기능 paramter를 다 보여 준다고 한다. 그 밖에, NEC는 스탠포드와 2년 계약을 맺고 연구원을 파견해 Openflow 라우터를 만들고 있다.

(변) NetFPGA는 연구자가 직접 건드릴 수 있는 hardware-forwarding router/switch 를 주는 것이 의미있다. s/w와 h/w간 interface, forwarding engine의 common 부분이 완성되어, 연구자가 자신의 idea를 적은 노력으로 직접 h/w로 구현할 수 있다는 것은 큰 장점이다.

Routing protocol 이상 상위 layer하는 사람은 하드웨어는 안 건드리고 FIB table에 소프트웨어로 write만 하면 되고, 특별한 packet processing 까지 하고 싶으면 그 부분만 verilog로 coding 해서 구현하고, 아무튼 이렇게 \$499짜리 NetFPGA 보드를 PC에 설치하면 마음대로 고쳐볼 수 있는 고속 포워딩 되는 4port GE router 를 갖게 된다. 국내 학계에서도 적극 활용하면 좋겠다.

NetFPGA가 partitioning되는 virtualization까지 지원하기 위해선 상당히 쉽지 않은 작업까지 해야 할 것이고, OpenFlow 스위치로서의 역할은 쉽게 될 것이다.

소스 공개 정책은 미국의 거의 모든 testbed에서의 방향이다. PlanetLab도 전체를 공개하고 있고, GENI prototyping project들도 h/w 및 s/w 모두 공개하겠다는 동의서에 서명해야 한다.

(김) 맥퀸 교수가 말하는 0-byte queue라는 것도 재미있다. Queue(또는 memory)가 크면 packet swiching을 구현할 수 있는 것이고, 그 크기를 0로 가져 간다는 것은 결국 flow switch, circuit switch를 만드는 것이니 스위치 레벨(Layer 1?)에서 부터 자기 나름대로의 아이디어를 구현해 보고 싶은 사람들에게는 더 없이 좋다.

대부분 소프트웨어로 한다는 것도 결국 Routing & Forwarding(or Switching)을 구별해 생각할 때에 Control에 해당하는 Routing algorithm 등은 소프트웨어로 처리하고 (그래서 versatile하게, 여러 다양한 라우팅 방법 시험해 볼 수 있게), 그 계산 결과에 따른 forwarding/switching execution은 PGA h/w로 함으로써 throughput을 multi-giga bps로 올리는 장점이 있다.

결국 NetFPGA는 virtualization을 어느 하위 계층까지 원하느냐에 따라 편리한 틀로 생각할 수도 있고 불편한 틀로 생각할 수도 있다.

(박) 말씀하신대로, NetFPGA가 완벽한 solution을 제공하기 보다는, 이제부터 연구자들의 아이디어를 모으자 라는 분위기이다. 이런 목

적을 위해 전략적으로 모든 소스를 공개하는 것 아닌가 생각된다. 또, 이번 Sigcomm 발표자들이 Future work 를 언급하는 부분에서 NetFPGA를 이용한 확장을 많이 언급하는 것으로 볼 때, 이런 Trend가 계속된다면 NetFPGA를 근간으로 하는 Utility나 Verilog component 들이 향후 많이 나올 수도 있을 것 같다.

### 3. 소프트웨어에 기반한 네트워크 가상화

(문) 이번 ACM SIGCOMM PRESTO 워크샵에서 Intel Research는 Software & PC-based architecture도 scalable하다고 발표하였다. NetFPGA를 굳이 쓰지 않아도 고성능 PC로 어지간한 성능은 구현해 볼 수 있다는 생각이 들었다. 영국 UCL의 Mark Handley 팀에서는 Virtual Router로 Xen Hypervisor 위에서 돌아가는 VR 성능을 발표했다. VR의 core components는 Eddie Kohler의 Click이다. 고성능 PC의 가능성을 극대화한 예라고 생각된다.

### 4. 앞으로의 방향

(박) NEC연구원에게 "Openflow가 상업화 되려면 10년이 걸릴 수도 있는데 commercial 회사가 그런거 연구해도 되냐" 라고 물었더니, "우리(NEC)는 길게 보고 연구한다. 10년후에 상용화 될 연구를 위해 지금 자금 투입하는 건 당연하다."라고 답했다. 우리나라와 비교해보며, 자세의 차이가 결과의 차이를 만들 수도 있겠다는 생각을 한 순간이었다..

(김) 이처럼 다양한 새 기술의 global deployment /operation /test 를 하는 것이 중요하다고 생각되는 것이다. 물론 이러한 환경을 우리 자체가 만들어 나가는 것도 해 볼 일이지만, 우선은 이렇게 매우 매력적인 방법과 그룹이 형성되고 있다면 그 활동에 신속하고 적극적으로 동참해 국제협력 하에 우리의 연구를 진행해 나가는 것도 좋겠다. 환경 기술은 아니더라도 핵심 프로토콜 기술에서 참신한 것 만들어 낼 수 있다.

(정) PlanetLab의 virtualization/slicing/federation 이 최근 나타나는 서비스 특성들 그리고 서비스 방향 등을 고려할 때, 얼마나 적절히 User Requirements 및 Business Trends 그리고 하부 전달망의 기능 및 구조 등과 조화를 이루어 능동적으로 적용할 수 있는 구조인가 생각해보자. 당분간 네트워크의 추세가 Transport 제어 기능 및 서비스 제어 기능이 구분되어 signaling 메커니즘에 의하여 FMC가 이룩되는 QPS 환경으로 발전될 것으로 보이는 데, 이러한 구조를 얼마나 잘 적용시킬 수 있을지 모르겠다. 매우 가까운 시일 안에 Telco/인터넷 서비스 구분이 없어지는 것이 분명한데, 서비스 비즈니스 활성화 측면에서 우리나라 미래를 위하여 PlanetLab의 과정보다는 PlanetLab++ 등의 우리 특성에 맞는 구조의 새로운 선택/집중은 필요하지 않은가 궁금하다. 이러한 문제에 대하여 검토된 사항이 있다면 논의해보면 좋겠다. 이에 따른 Transport 네트워크 장비의 개발 구조도 미래 지향적인 구조로 설정되어야 한다.

(이) 미래 인터넷 테스트베드 관점에서 보면, 미래 인터넷에서는 Global Inter-network 를 먼저 추구하고 있기에 실험을 위한 Testbed 는 기존의 연구 결과물들을 최대한 활용하여 먼저 꾸며보는 것이 좋을 것 같다. 그렇다고 이것이 반드시 화려해야 할 필요성이 있나 하는 의문이 다. KOREN 이나 KOREN 의 upgrading 위에 VM 기능이 있는 서버들을 붙여서 기본적인 Testbed 를 충분히 구축할 수 있을 것으로 본다. 그 보다는 중요한 것이 Inter-network 이전에 우리 나름의 아키텍처로 Intra-Network 를 먼저 시도해보는 노력이 훨씬 좋지 않나 생각된다.

(박) ACM SIGCOMM PRESTO 에서도 미래 인터넷 설계에 대한 아래와 같은 많은 질문들이 나왔다. Router 를 어떻게 더 reliable 하게 설계할 수 있느냐가 어려운 문제다. (Albert Greenberg) 30년 전 main frame 에서는 어떻게 이 문제를 해결했는지 다시 살펴봐야한다 (Paul Gleichauf). Router configuration 을 쉽게 할 수 있는 방법을 개발해서 다양한 configuration 을 쉽게 실험해 볼 수 있으면 큰 도움이 될 것이다. (Bruce Davie). Cloud computing market 에서는 쉽게 하는 일이 Enterprise network market 에서는 아직 잘 안 되고 있다. (Albert Greenberg). 우리가 필요로 하는 게 Xen 수준의 가상화인가, 아니면 PlanetLab 수준인가? (Jeff Mogul) PlanetLab 에서는 스케줄링을 원하는대로 할 수가 없어서 네트워크 layer 에서는 Xen 수준이라고 생각된다 (Michael Freeman)

(박, 문) 네트워크 가상화는 하위 계층에서 상위 계층까지 vertical하게 자원을 공유할 수 있는 기능을 제공해준다. 지금까지의 dark fiber lease와는 전혀 다른 개념의 서비스가 생겨날 수 있다. 본 기고문은 이에 대한 정답을 제시하기 보다는 우리가 생각하고 있는 방향과 아이디어를 정리해보자는 취지로 편집되었다. 앞으로도 이러한 토론이 국내 학계 및 업계에서 활발하게 진행되길 기대해본다.