

ISBN 978-89-6211-267-2

## UCLP 프로젝트의 변화방향 조사/분석

일자: 2008년 11월 14일

부서: CNI사업단 연구망개발팀

제출자: 홍원택, 공정욱

{wthong, kju}@kisti.re.kr



**한국과학기술정보연구원**  
Korea Institute of Science and Technology Information

305-806 대전광역시 유성구 어은동 52번지  
TEL (042)869-0676 / FAX (042)869-0679  
www.kisti.re.kr

# UCLP 프로젝트의 변화방향 조사/분석

작성자: 홍원택, 공정욱

연구망개발팀, CNI사업단, 한국과학기술정보연구원  
{wthong, kju}@kisti.re.kr

최종수정일: 2008년 11월 14일

## Abstract

UCLP 프로젝트는 네트워크 자원의 소프트웨어적인 표현 및 APN(Articulated Private Network)이라는 사용자 제어 가능한 사설 네트워크 개념을 탄생시켰다. 최근, UCLPv2의 후속 연구 결과물로 P2V(Physical to Virtual) 개념을 가능하게 하는 IaaS(Infrastructure as a Service) 프레임워크가 새롭게 제안되었고, 네트워크 각 계층에서 IaaS 기반의 세부 프로젝트들이 연구되고 있다. 본 문서에서는 이전의 UCLP 프로젝트 및 UCLP를 이용하여 연구망에서 수행되었던 시연들을 간략히 정리하고, IaaS 프레임워크 및 관련 프로젝트들(Argia, MANTICORE, Ether, GRIM)을 조사 및 분석한다.

## Topics

1. 서론
2. UCLP 개요
3. UCLP 관련 데모
4. IaaS (Infrastructure as a Service)
5. IaaS 프레임워크 기반의 프로젝트
6. 결론
7. 참고문헌
8. 부록

## 1. 서론

독점적인 네트워크 사용의 필요성이 증대되면서 람다 네트워킹 기술이 연구 및 교육 망 분야에서 활발히 사용되는 추세이다. 람다 네트워킹을 통해 종단간 대용량 데이터를 전송하려는 그리드, e-Science 응용 연구자들은 고 성능, 저 지연의 특성을 갖는 독점적인 대역을 제공 받을 수 있다. 일반적 관점에서의 람다 네트워킹의 목적은 특정 목적을 같이 하는 연구 그룹들이 그들 자신의 "lightpath"를 제공 받는 것이다. 이러한 "lightpath"를 통해 연구자들은 고 대역의 응용에 특화된 독점적인 네트워크를 구성할 수 있게 된다.

람다 네트워킹을 효율적으로 제공하기 위해 개발된 User Controlled LightPath (UCLP)는 Web Services 기능과 향상된 유저 인터페이스를 지원해 여러 종류의 장비들을 대상으로 람다 네트워킹을 지원할 수 있는 확장성을 제공한다. 현재까지 KREONet2 및 GLORIAD-KR 환경에 맞게 UCLP 시스템이 확장 개발 및 구축되었고, 국내외 데모에 성공하였다. 한편, UCLP 프로젝트는 Ver2.에 이어 IaaS(Infrastructure as a Service)라는 새로운 프레임워크를 기반으로 계층별 특성에 맞는 세부 프로젝트들로 변화를 거듭하고 있다. 본 문서에서는 UCLP 프로젝트를 간략히 소개하고, IaaS 프레임워크 및 관련 프로젝트들을 조사 및 분석하여 향후 발전 방향을 모색한다.

## 2. UCLP 개요

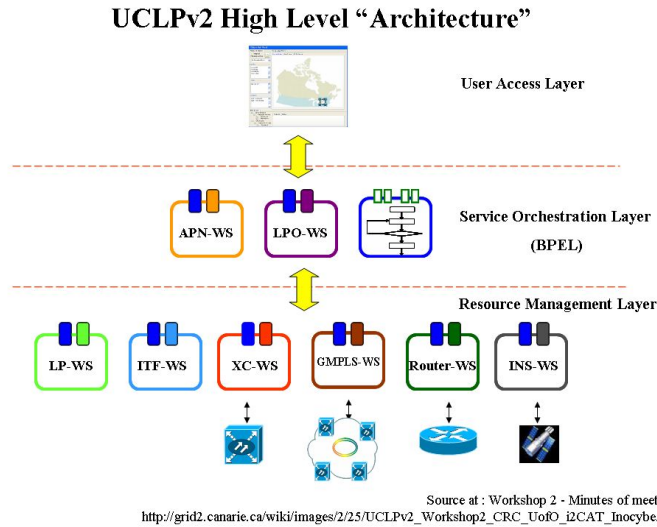
Solana Networks, UQAM - University of Ottawa, 그리고 University of Ottawa - CRC - i2CAT - Inocybe Technologies (CIU)의 세 팀에서 각기 다른 방법으로 UCLPv2 를 구현하였고, 이 중 가장 완성도가 높은 CIU의 UCLPv2가 가장 보편적으로 선택되어 적용되었다. UCLPv2의 다음 버전도 CIU의 버전을 기초로 하여 발전되어 오고 있다.

### 2.1 UCLPv2의 구조

UCLPv2 는 물리적인 네트워크 자원을 웹서비스를 이용하여 가상의 자원으로 만들어서, 이들을 사용자가 조합하여 원하는 네트워크를 만드는 것을 돕기 위한 구조를 가지고 있다. UCLPv2 의 구조는 크게 다음과 같이 세 개의 레이어로 구성된다.

- 자원 관리 레이어 (Resource Management Layer) : 물리적인 네트워크들로부터 생성된 웹서비스가 가능한 자원들로 이뤄진다.
- 서비스 조합 레이어 (Service Orchestration Layer) : 웹서비스들로 이루어진 가상네트워크 자원들을 조합하여 사용자가 네트워크 자원 할당하는 일을 도와준다.
- 사용자 접근 레이어 (User Access Layer) : Graphic User Interface (GUI)를 통하여 사용자(사람)가 하위 레이어에 있는 모든 웹서비스들을 접근할 수

있게 한다. 웹서비스 인터페이스를 통하여 어플리케이션 프로그램들도 하위 레이어에 있는 웹서비스들을 접근할 수 있다.



<그림 1> UCLPv2 High Level Architecture

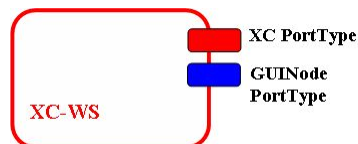
다음은 여러 웹서비스들에 대하여 각각의 역할과 구현방법, 그리고 이용자에 대하여 소개한다.

- NE-WS (Network Element Web Service)
  - 통신 장치를 제어하기 위한 웹서비스들의 집합
  - Axis 를 이용한 Web Service
  - 제어하고자 하는 통신 장치에 따라 다른 종류의 웹서비스를 제공
  - 회선(망) 제공자 측에 설치
  
- LP-WS (LightPath Web Service)
  - 하나 또는 그 이상의 연결된 자원들 사이의 하나의 링크를 추상화
  - 하나의 BPEL (Business Process Execution Language) 워크플로우
  - LP의 종단점에는 네트워크가 가능한 어떤 것도 될 수 있다.
  - 회선(망) 제공자 측에 설치
  
- ITF-WS (InTerFace Web Service)
  - 하나의 네트워크 장치에 있는 하나의 자원을 표현하는 BPEL 워크플로우
  - 회선(망) 제공자 측에 설치
  
- APN-WS (Articulated Private Network Web Service)

- 호스트 서버 측에 여러 워크플로우들로 이루어진 웹서비스들을 포함하며, APN 자원 목록과 다른 자원들로부터 여러 웹서비스들 함께 연결하는 워크플로우 스크립트
  - APN 제공자 측에 설치
- LPO-WS (LightPath Object Web Service)
- 회선(망) 제공자 측에 설치하는 것을 제외하고는 APN-WS와 동일
  - APN 사용자들은 LPO-WS를 분해하거나, 내부의 각각 자원에 접근 할 수 없다.
  - 회선(망) 제공자 측에 설치

## 2.2 UCLPv2의 주요 Web Service

NE-WS 의 예로 다음과 같은 XC-WS 가 있다. XC-WS 는 광통신 스위치에 Cross Connect (교차연결) 에 관련된 명령을 실행할 수 있다. XC PortType 란 앞서 말한 교차연결에 관한 명령들의 집합이고, GUINode PortType 란 사용자가 쉽게 통신장치를 제어하기 위한 여러 정보들을 제공하기 위한 명령들의 집합이다. 이를 이용하여 사용자에게 GUI를 통한 쉬운 사용자 인터페이스를 제공할 수 있다. NE-WS 의 예로, SONET/SDH, Fibre, Lambda Cross Connection 자원을 다루는 XC-WS (Cross Connect Web Service) 가 대표적이며, VLAN 기능을 지원하는 이더넷 스위치에 대한 802.1q-WS, GMPLS 그룹에 대한 GMPLS-WS, MPLS 그룹에 대한 MPLS-WS, 그리고 레이어 3 라우터에 대한 Router-WS를 생각할 수 있다.



<그림 2> NE-WS의 예(XC-WS)

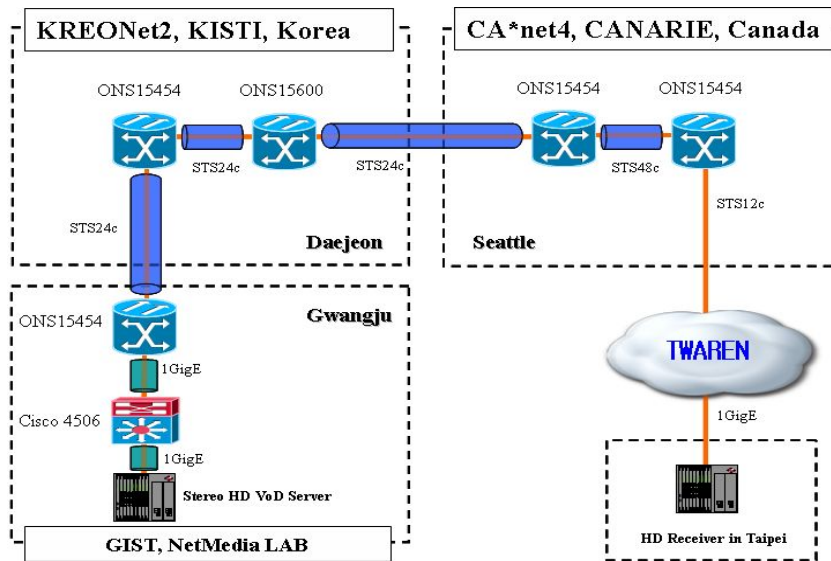
LP-WS (LightPath Web Service) 을 이용하여 사용자 접근 레이어에 추상화 레이어를 제공할 수 있다. 사용자들이 실제 물리적인 광통신 스위치 및 여러 장치의 세부 사항을 몰라도, LP-WS를 통하여 광 패스를 이용할 수 있게 추상화된 레이어를 제공한다. 이러한 LP-WS는 웹서비스 워크플로우로 이루어져 있으며, BPEL 로 구현한다. UCLPv2 는 BPEL 을 이용하여 컴파일 없이 웹서비스들을 추가하고 삭제하는 과정을 할 수 있다. 따라서 다양한 자원들의 연결을 표현하는 LP-WS 에 동적으로 자원을 추가하고 삭제하는 일이 가능하다. 이와 같이 웹서비스의 워크플로우를 이용하여 사용자에게 다양한 링크를 제어할 수 있으며, 사용자에게 세부 물리적인 네트워크 장치 및 구성을 감출 수 있다.

여러 LP-WS 의 집합으로 네트워크를 이룰 수 있으며, 이를 APN-WS (Articulated Private Network - WS) 이라고 한다. APN-WS 는 동일한 물리적인 네트워크 자원을 이용하여 여러 다양한 가상의 네트워크를 제공한다. 특히, APN-WS 는 LP-WS, NE-WS 등 다양한 웹서비스들을 함께 사용자에게 제공하여, 사용자가 원하는 대로 네트워크 생성하고 운영할 수 있게 한다. APN 은 동일한 네트워크 설정을 가지고 있으며, 한번 설정을 한 후에는 토폴로지와 대역폭이 고정된다. 이 후 설정을 해제하면 다른 사용자가 이용할 수 있다. 이와 같이 APN-WS 는 사용자가 직접 설정하고 제어할 수 있는 네트워크를 제공한다.

웹서비스의 사용자는 (물리적인) 네트워크 관리자, APN 관리자, 그리고 사용자로 구분할 수 있다. 네트워크 관리자는 일반적으로 회선(망) 제공자들이라 생각할 수 있고, APN 들이 이용하기 위한 네트워크 자원 할당과 네트워크 자원들을 생성하는 역할을 한다. 즉, NE-WS, LP-WS, 그리고 ITF-WS 를 생성하고 제공하는 역할을 한다. 그리고 LPO-WS 를 구성하여 특정 네트워크 서비스 제공자 (예, ISP) 들에게 제공하는 역할을 한다. APN 관리자는 네트워크 관리자 또는 다른 APN 관리자들로 부터 APN 자원들을 할당받아 사용자들을 위하여 APN 네트워크 설정하는 역할을 한다. APN 관리자들은 자신들이 할당받은 자원을 분배하거나 합치거나 할 수 있으며, 다른 APN 관리자들에게 빌려줄 수 있다. 사용자는 실제 웹서비스를 이용하여 네트워크를 사용하는 사람들을 뜻한다. APN 관리자들이 제공하는 APN 설정들을 이용하여 주어진 네트워크를 이용할 수 있다.

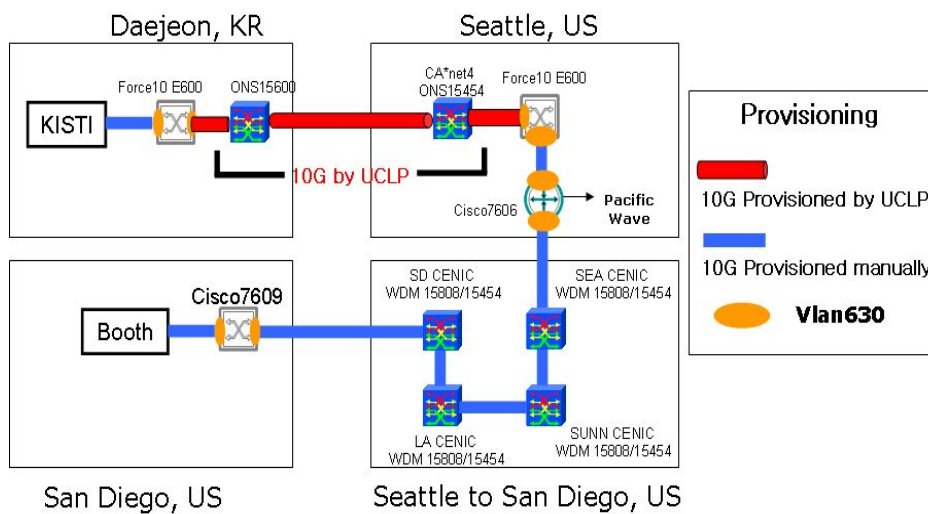
### 3. UCLP 관련 데모

2005년 8월 26일 타이완의 타이페이에서 20th APAN meeting때, UCLP demonstration이 있었다. 아래와 같이 광주에서 미국의 시애틀을 거쳐 타이페이까지 HDTV traffic을 전송하기 위하여 광주에서 시애틀까지의 lightpath를 UCLP로 설정하고 해제하였다. 추가적으로 시애틀부터 행사장까지의 lightpath는 타이완 측에서 구축된 UCLP를 이용하여 설정 및 해제되었고, 부분적으로는 Vlan 기술을 이용한 관리자의 수동 설정을 통하여 데모를 수행하였다. 이 데모는 UCLPv1을 기반으로 수행되었고, 데모를 통해 Layer 2에서의 Vlan 기술을 UCLP에서 지원해야 할 필요성이 제기되었다. 이후, Force10 장비를 대상으로 부분적인 Vlan 지원을 할 수 있도록 추가 개발이 이루어졌다. 개발의 한계점은 SONET layer에서의 프로비저닝과 Vlan을 대상으로 한 프로비저닝이 분리되어 수행된다는 점이었고, 향후 UCLPv2를 대상으로 한 확장에서는 이러한 부분을 수정하여 한 번에 프로비저닝이 가능하도록 변경되었다.



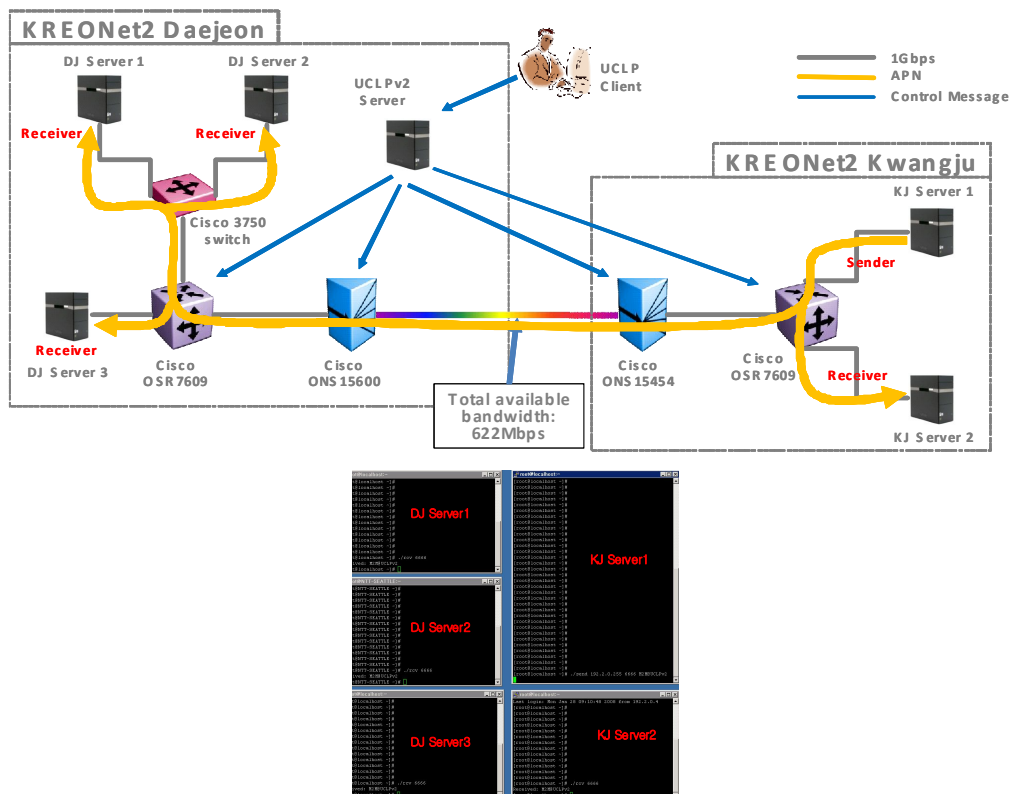
<그림 3> UCLP demonstration in APAN

또한, 2005년 9월 26일 미국의 샌디에고에서 iGrid2005 Workshop에서 UCLP Demonstration이 있었다. 대전에서 샌디에고까지 10Gbps lightpath중 대전에서 시애틀까지의 lightpath를 할당하였다. 이렇게 구축된 독점적인 경로를 통해, Uncompressed HD 및 고에너지 물리 어플리케이션의 데이터가 안정적으로 전송되었다. 데모를 수행하는 과정에서 UCLP에서 보내는 제어를 위한 경로를 확보하지 않아 Lightpath 설정 및 해제 과정에서 문제를 발생시켰었다. 각 NOC 관계자들의 빠른 대응으로 문제를 빨리 해결했었지만, 시연시 백업 링크 확보의 중요성을 다시 한 번 확인하였다.



<그림 4> UCLP demonstration in iGrid2005

높은 완성도에도 불구하고 UCLPv2는 Layer 1의 SONET/SDH 기술에 기반한 link-level APN만을 구성함으로써, 다대다 연결이 필요한 HDV와 같은 응용에서는 복수개의 중단간 연결을 맺어야 하는 한계가 있었다. 이러한 한계점을 해결하기 위해 Layer 2의 Vlan 기술을 기반으로 Network-level APN의 구성이 가능하도록 기능을 향상시켰었고, 다음 그림은 대전~광주 간에서 Network-level APN을 구성하고 멀티캐스트 메시지를 뿌려 수신측에서 동시에 전송받음으로써, 좀 더 다양한 네트워크 응용들을 수용할 수 있도록 하였다. 2008년도 APAN 및 ISC 행사에서도 실험 네트워크를 기반으로 Network-level APN 구성을 위한 데모가 수행되었다.



<그림 5> 대전~광주 간 Network-level APN 구성 시연

#### 4. IaaS (Infrastructure as a Service)

본 절에서는 UCLPv2에서 진화되어 물리적인 자원을 일반화시킬 수 있는 접근 방법인 IaaS(Infrastructure as a Service) 프레임워크를 살펴본다.

##### 4.1 IaaS에서의 가상화(Virtualization)

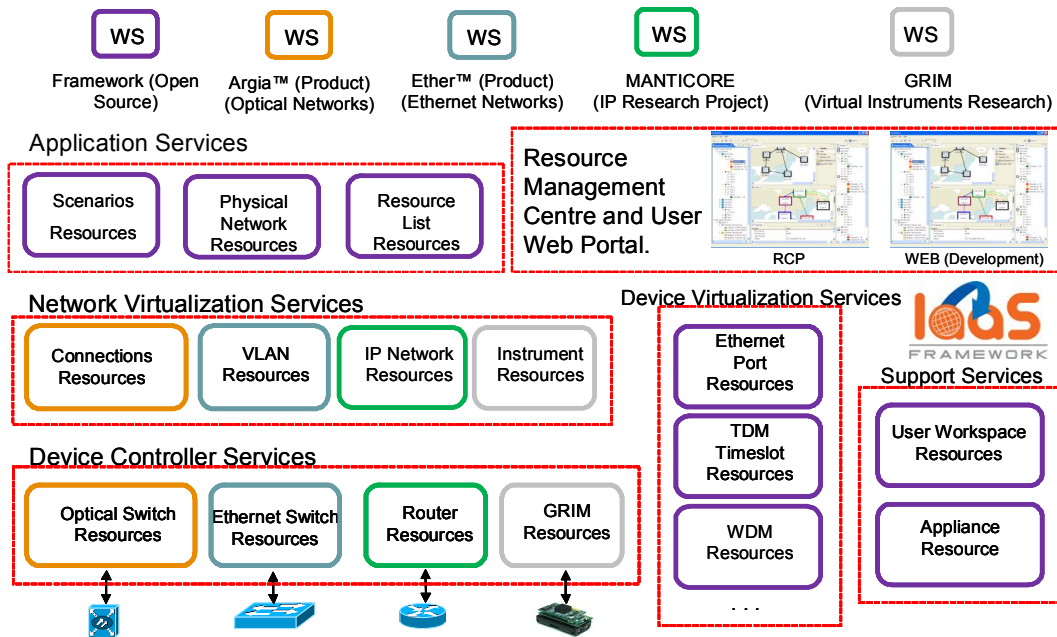
가상화는 현재 시스템 및 네트워킹 분야에서 많이 논의되고 있는 이슈이다. 실질적으로 "virtualization"은 각 분야의 사람들에게 다른 의미를 갖는다. 그러므로, IaaS 프레임워크의 문맥에서 가상화가 무엇을 의미하는지 정의하는 것이 필요하다. IaaS에서의 가상화는 물리적인 장비를 소프트웨어 객체(예를 들어, Web service 또는



Web service 자원과 같은 객체)로 표현하는 기술을 의미한다. 이러한 가상화 기술은 PC에 폭넓게 적용된다. VMWare, Xen과 같은 툴들은 동일한 하드웨어 플랫폼을 기반으로 여러 개의 가상 머신들을 제공한다. 본래 서버 사용을 극대화하여 데이터 센터 비용을 줄이는 수단으로 사용되었던 PC 가상화 기법은 Amazon과 같이 독자적인 솔루션을 사용해서 하드웨어를 렌트해 주는 회사들에 의해 확장되었다. IaaS는 SaaS (Software as a Service)와 동등한 개념으로서 사용자는 장비를 직접 구입하지 않고, 대신 사용료를 브로커를 통해 지불하는 개념이다. 서비스를 제공하는 동안 사용자는 실제 장비의 소유주처럼 인프라를 사용하고 제어할 수 있다. 이러한 비즈니스 모델은 on-demand 서비스와 비교될 때, 보다 긴 시간의 자원 사용에 초점을 둔다.

IaaS 프레임워크는 CANARIE UCLP 프로젝트에서 수행된 수년간의 연구 결과물에 대한 일반화된 접근 방법이다. 2001년부터 두가지 버전의 UCLP 프로젝트가 광 네트워크에 대한 가상화 방법을 제공하기 위해 시작되었다. 초기 버전의 UCLP 목적이 도메인 간에 걸쳐 중단간 경로를 제공하는 것이었다면, UCLPv2는 재사용가능하고 설정 가능한 네트워크들을 생성하는 것을 목적으로 하고 있다. 이러한 UCLPv2 개념은 IaaS 프레임워크 기반의 다른 P2V(Physical to Virtual) 제품 및 연구 과제로 진화하였다. Argia는 광 네트워크(TDM, WDM, 광기술)를 위한 제품으로, Ether는 Ethernet과 MPLS 네트워크를 위한 제품으로, MANTICORE는 논리적으로 분리된 IP 네트워크를 위한 연구 프로젝트로, GRIM은 센서 및 기타 장비들의 가상화를 적용하는 연구로 수행되고 있다.

IaaS 프레임워크는 아파치 소프트웨어 라이선스 v2 기반의 자원, 라이브러리, 툴들의 집합으로써 개발자들이 프레임워크 프로그래밍 모델 기반의 새로운 가상화 솔루션의 생성을 가능하게 해준다. 이러한 툴에 의해 제공되는 기능들은 개발자가 물리적인 인프라를 서비스로(지원되는 SOAP 엔진들은 Axis2, CXF, Spring-WS등을 포함한다.) 나타내는데 사용될 Web service 스택을 선택하게 해주고, 보안, 예약관리, 인프라 서비스에 대한 데이터 지속성과 같은 기능 등을 플러그인하는 일련의 모듈들을 제공한다. 또한 프레임워크는 프로토콜 파서(TL1, Netconf)와 같이 장비와 통신하는 드라이버 개발을 가속화 시키는 라이브러리들을 제공하고, 전송 핸들러(TCP, SSL, SSH), IaaS 엔진으로 불리는 드라이버 구조를 제공한다.



<그림 6> IaaS 프레임워크의 구조 및 관련 프로젝트

그림 6은 IaaS 프레임워크의 구조 및 관련 프로젝트들을 보여준다.

Device Controller Services는 프로토콜을 사용하여 장비와 통신하는 장비 컨트롤러 서비스들과 장비를 표현하는 새로운 자원을 생성하여 가상화를 제공하는 가상화 팩토리 서비스들의 집합이다. (예를 들어, SONET 스위치 포트에 있는 TDM 채널들의 집합이나 라우터의 Ethernet 인터페이스)

“가상화”의 결과로 생성된 자원들은 Device Virtualization Services 그룹에 속한다. 이러한 서비스들은 장비의 일부를 표현한다. 즉, Ethernet Port 자원은 Ethernet 인터페이스를, TDM Timeslot 자원은 TDM 인터페이스에서의 채널들의 그룹을, WDM 자원은 WDM 인터페이스에서의 파장을 나타낸다. 이러한 자원들은 인프라를 서비스로 제공하기 위해 기관들에 의해 서로 교환될 수 있다. 예를 들어, A 기관이 8개의 TDM 인터페이스와 16개의 Ethernet 포트들이 있는 SONET 스위치를 보유한다면, A 기관의 관리자는 그 중 4개의 TDM 인터페이스와 8개의 Ethernet 포트를 가상화 시킬 수 있고, 그것들을 B 기관에게 줄 수 있다. 그러면, B 기관은 넘겨받은 포트들을 실제 소유주처럼 제어할 수 있다. 더 나아가, A 기관이 다른 4개의 TDM 인터페이스와 8개의 Ethernet 포트들을 가상화 시킬 수 있고, 그것들을 C 기관에게 줄 수 있다. 이러한 경우, B와 C 기관은 동일한 광 스위치의 일부를 실제 소유주처럼 제어할 수 있다.

Network Virtualization Services는 종단 사용자에게 가상 인프라(서비스로서의 인

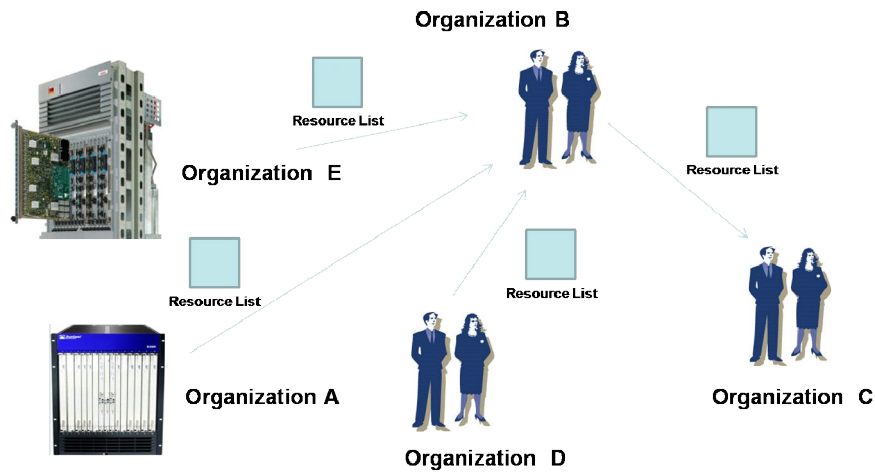
프라)를 제공하는 서비스들이다. Network virtualization services들로는 사용자가 TDM, WDM, 광 자원들 상에서 중단간 연결을 수행하도록 하는 Argia의 optical connection service, MANTICORE의 IP Network service들을 예로 들 수 있다.

## 4.2 자원 트레이딩 모델

Resource Lists Services는 기관들 간에 자원들을 서로 교환하는 수단을 제공한다. A 기관이 자원들 중의 일부에 대한 접근 퍼미션을 B 기관에게 주기를 원할 때, A 기관은 접근할 수 있는 물리적인 인프라를 표현하는 장비를 가상화한 자원들로 활성화된 자원 리스트를 생성하고, B 기관에게 그것을 준다. B 기관이 자원 리스트를 받으면, B 기관은 디플로이했던 Network virtualization services에게 device virtualization resources을 할당할 수 있다. (예를들면, 4개의 Ethernet 포트들은 IP Network service에 의해, 8개의 TDM 포트들은 Optical Connection Service에 의해 사용될 수 있다.) 이러한 자원 교환 프로세스는 그림 7에서 보여지는 것처럼 재귀적이다. 즉, B기관은 다른 자원 리스트를 생성할 수 있고, B 기관이 임시로 소유한 자원들 중의 일부에 대한 액세스를 다른 기관에게 줄 수 있다. 향후 개선사항은 기관이 <http://www.vinfrastructures.com>과 같은 알려진 자원 브로커 사이트로 자원 리스트를 공개하는 것이다. 그러면, 관심있는 기관들은 이러한 사이트를 브라우징할 수 있고, 그들이 필요로 하는 인프라 자원들을 얻을 수 있다.

기관들과 사용자들은 User Workspace Service에 의해 관리된다. 기관에 있는 각 사용자는 계정 및 관련된 증명을 갖고, 이러한 것들은 여러 서비스들을 접근할 때 사용될 것이다. 사용자들은 권한에 따라 3단계의 다른 역할을 수행할 수 있다.

- Physical infrastructure Admin: 물리 인프라의 소유주로서 소유한 물리 자원들을 가상화 시킬 수 있고, 자원 리스트를 생성하고 공개함으로써 다른 사용자들이 그것들을 제어할 수 있도록 퍼미션을 할당할 수 있다.
- Virtual infrastructure Admin: 이러한 범주의 사용자는 하나 이상의 물리 인프라 관리자 또는 다른 virtual infra 관리자로 부터 자원들을 얻는다. 또한 제어할 수 있는 자원들을 다른 network virtualization services에 할당할 수 있다.
- End User: 이러한 범주의 사용자는 중단간 connection service나 IP network service와 같은 서비스를 사용하기 원하는 사용자이다.



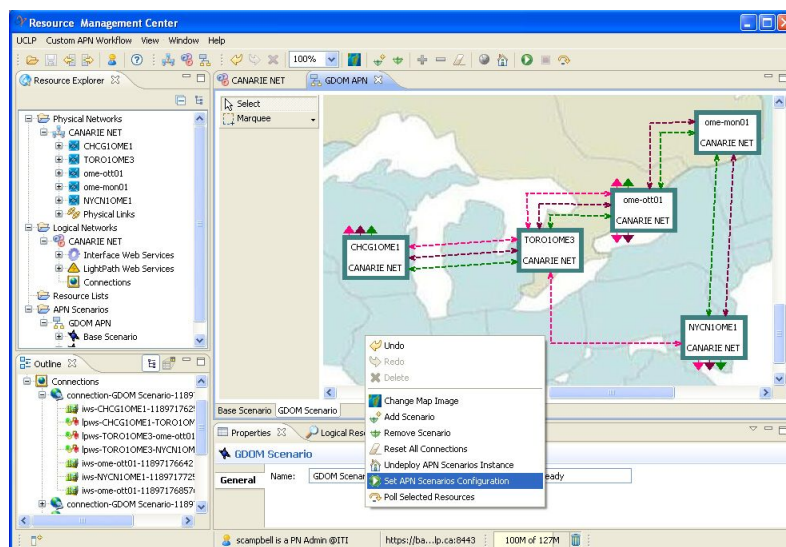
<그림 7> IaaS 프레임워크에 의해 활성화된 자원 트레이딩 모델

## 5. IaaS 프레임워크 기반의 프로젝트

본 절에서는 IaaS를 기반으로 하는 여러 프로젝트들을 소개한다.

### 5.1 Argia

Argia는 광 네트워크를 대상으로 하는 실질적인 UCLPv2의 차기 버전이다. 대부분의 UCLPv2 기능을 포함하여 설계되었다. P2V 개념을 이용하여 광 네트워크 자원들을 자원 리스트로 구성한 후, 종단 사용자를 위한 "APN Scenarios"를 구성할 수 있고, 사용자들은 "predefined"된 시나리오를 호출하여 원하는 네트워크 응용을 실행할 수 있다. 또한 Argia는 광 멀티캐스트를 지원하여 복잡한 멀티캐스트 SONET/SDH 토폴로지를 생성할 수 있다.

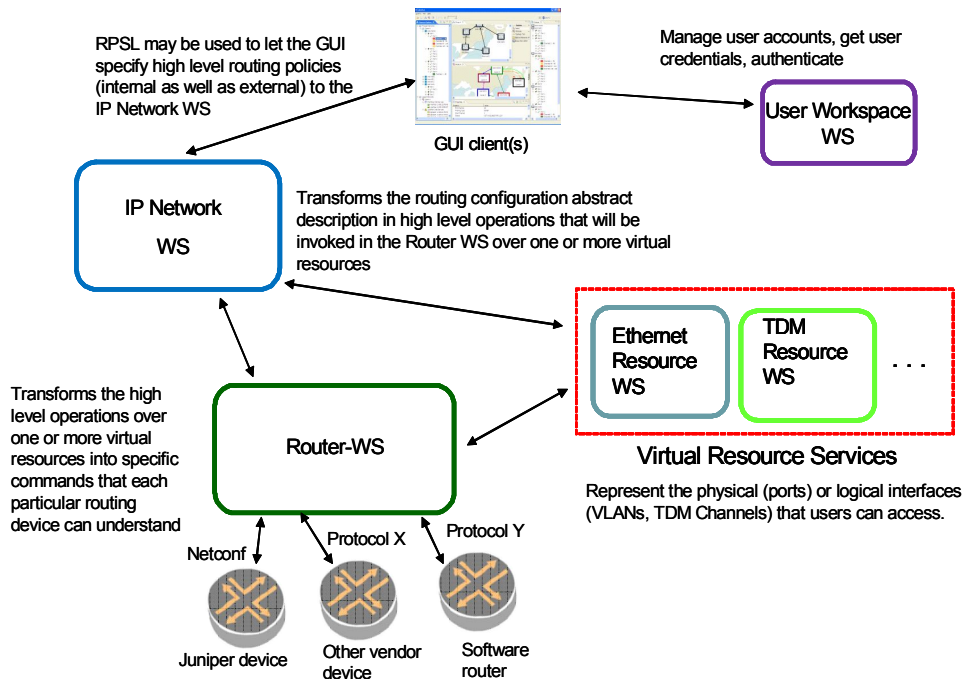


<그림 8> Argia에서의 Optical Network P2V

현재, Argia는 많은 데모에서 사용되어 오고 있다. 예를 들어, California와 Canada Ottawa 지역 간의 "Collaborative Architecture Design"을 위한 Eucalyptus라는 프로젝트에서 협업 환경을 제공하기 위한 하부 기술로 사용되었다. 또한, 여러 사이트에서 동시(mp-to-mp) 고성능/고대역/고화질의 디지털 미디어 스트리밍을 요구하는 HPDM(High Performance Digital Media) research initiative에서도 활발히 사용되고 있다.

## 5.2 MANTICORE

MANTICORE는 논리적으로 분리된 IP 네트워크를 위한 연구 프로젝트로 주요 소프트웨어 모듈은 Router WS(device controller service), IP Network WS(network virtualization service)이다.



<그림 9> MANTICORE 소프트웨어 구조

### 5.2.1 Router WS

Router WS는 하나 이상의 라우터 장비와 통신하는 WSRF 기반의 Web Service이다. 물리/논리 인터페이스, 논리 라우터, 알람, 설정을 포함하는 라우팅 장비의 상태는 WS Resource로 공개되고, Router WS의 클라이언트들은 자원 속성을 쿼리함으로써 물리 장비의 상태를 접근할 수 있다. Router WS 인터페이스는 라우팅 장비의 기능을 포함하는 high level 오퍼레이션들을 제공한다. 예를 들어, "invoke" 명령을 사용하면서 오퍼레이션 ID와 파라미터들을 넘겨준다면, Router WS 클라이언트는 인터페이스의 넷마스크를 설정하고, 논리 인터페이스를 생성/제거하고, 정적 경로를

삽입/제거하며, RIP, OSPF, BGP와 같은 동적 라우팅 프로토콜을 설정할 수 있다. 오퍼레이션들이 요구하는 파라미터들은 특정 라우팅 장비만을 위한 것일 수 없고, 파라미터들은 이상적으로 모든 라우터 장비들에 대해 적용될 수 있다는 점이다. 예로서, 라우터 인터페이스에 OSPF를 설정하는 오퍼레이션은 Juniper 장비가 필요한 파라미터뿐만 아니라 Cisco 라우터가 필요한 파라미터들과도 동작할 수 있어야 한다.

Router WS는 추상화된 명령들을 각 라우팅 장비가 이해할 수 있는 메시지로 변환시킨다. 이러한 부분은 IaaS 엔진에 의해 수행된다. IaaS 엔진은 Router WS에 의해 사용되는 추상화 라우터 모델의 오퍼레이션들과 정보들을 네트워크 장비가 이해할 수 있는 프로토콜과 파라미터들로 변화시키는 일종의 드라이버 역할을 한다. 초기 버전의 MANTICORE 구현물은 NetConf API를 통해 설정되는 Juniper 라우터들을 위한 드라이버를 제공한다. NetConf는 소프트웨어 에이전트가 설정 메시지를 네트워크 장비에 보내도록 설계된 장비 독립적인 XML 프로토콜로써 일반적인 CLI (Command Line Interface) 방식 보다 좀 더 구조화된 방식이다.

Physical infrastructure 관리자는 Router WS "virtualize" 오퍼레이션을 호출하여 Ethernet Resource WS를 생성한다. 이 서비스는 라우터의 Ethernet 물리/논리 인터페이스를 나타내고, 이후 자원 리스트에 삽입되어 다른 기관에 공개될 수 있다. 결과적으로 해당 기관의 사용자들은 물리 인프라를 제어할 수 있게 된다.

### 5.2.2 IP Network WS

IP Network WS 또한 WSRF 기반의 Web Service로써 하나 이상의 IP Network Resources를 관리한다. 각 IP Network는 Ethernet resource services 집합을 가리키는 포인터를 갖는다. 이러한 Ethernet resource services는 IP Network이 제어할 수 있는 물리/논리 라우터들의 포트들을 표현한다. IP Network interface는 다음과 같은 기능들을 클라이언트들에게 제공한다.

- 라우터 인터페이스의 일반적인 설정. 사용자는 IP 주소, NETMASK, up/down을 포함하는 상태정보, 인터페이스의 일반적인 파라미터들을 설정할 수 있다.
- 정적 경로의 추가/삭제. 사용자는 정적인 경로 엔트리를 라우팅 테이블에 적용하고 이러한 엔트리는 하나 이상의 소스 인터페이스를 위한 것이 될 수 있다.
- 인터페이스나 네트워크 상의 IGP 설정. IP Network 사용자는 OSPF와 RIP와 같은 동적 라우팅 프로토콜을 모든 인터페이스에 적용할 수 있고, IGP 파라미터들을 변경시킬 수 있다.

- 인터페이스나 네트워크 상의 EGP 설정. 사용자는 피어링을 원하는 IP Networks을 결정하고, 외부 네트워크에 알려질 경로를 선택할 수 있다.

IP Network WS로 전송될 설정 데이터를 모델링하기 위해서는 간단하고 표현이 다양하며, 확장성이 있는 데이터 구조나 언어가 필요하다. 이러한 설정 데이터는 주로 IGP 설정이나 라우팅 정책 스펙 등에 적용되고, 장비 비의존적이어야 한다. 벤더 독립적인 형태로 IGP 설정뿐만 아니라 라우팅 정책들을 모델링하기에 적합한지를 알아보기 위해 RPSL(Routing Policy Specification Language) 언어가 선택되어 분석되었다. 결과적으로 IGP 설정들을 표현하기 위해 RPSL을 확장하는 것이 적합한 것으로 조사되었고, IP Network WS에서 모델링 언어로 사용될 것이다. 현재, RPSL 자바 모듈이 아직 구현되지 않은 상태이고, IP Network WS에 포함되지 않았다.

IP Network WS 라이프사이클은 다음과 같다. 일단 WS가 생성되면 접근할 수 있는 인터페이스를 나타내는 일련의 Ethernet Resource WS 집합으로의 포인터들을 포함하여 활성화 된다. IP Network WS 클라이언트는 필요한 오퍼레이션을 호출하고, RPSL과 같은 추상 언어로 설정 데이터를 전송한다. IP Network WS는 요구사항의 유효성을 검증하고, 설정되어야 하는 자원들을 리스트화 하고 접근이 가능한지를 확인한다. 그리고 Router WS들로의 요구 사항들을 생성한다. 오퍼레이션들이 성공적이면 IP Network WS는 Ethernet Resource WS의 자원 속성들을 업데이트한다. 사용자가 IP Network을 제거하고자 한다면, "destory call"이 IP Network WS에 불려지고, 특정 IP Network을 표현하는 자원이 제거된다.

### 5.3 Ether and GRIM

Ether는 Ethernet과 MPLS 네트워크를 대상으로, GRIM은 센서 및 기타 장비들의 가상화를 대상으로 프로젝트가 진행되고 있으나, 앞에서 언급된 프로젝트들에 비해 구체적인 설계나 구현물이 발표되지 않았다.

## 6. 결론

UCLPv1에서 "User-Controlled"라는 새로운 개념을 제안하여 사용자 측면에서 제어 가능한 Lightpath를 제공하던 UCLP 프로젝트는 UCLPv2를 거치면서 네트워크 자원을 소프트웨어 오브젝트로 표현함과 동시에 APN이라는 사용자 제어 가능한 일종의 사설 네트워크 개념을 탄생시켰다. 이후, UCLP 프로젝트는 기능상의 완성도를 높이면서, 네트워크 자원을 보다 일반화하여 다룰 수 있는 IaaS 프레임워크를 새롭게 제안하게 되었고, 추가적으로 관련 R&E 네트워크 커뮤니티는 Argia, MANTICORE, Ether, GRIM 등의 네트워크 각 계층에서 네트워크 자원의 가상화를 가능하게 하는 세부 프로젝트들을 진행하고 있다. 기존의 UCLP 프로젝트들과 달리 이후 일부의 세부 프로젝트들은 상용화를 목표로 진행하고 있다.

Argia, MANTICORE 프로젝트의 경우 완성도 높은 구현물이 발표되고 있는 상태이며, GENI, FIND와 같은 미래 인터넷 커뮤니티에서 주장하는 "Virtualization"의 개념을 제한된 범위에서 실현화시키는 측면에서 의의를 찾을 수 있을 것이고, HPDM과 같은 네트워크 응용과 연계한 대규모 시연에서도 채택되어 협업 환경을 제공하는 측면에서 많은 기여를 하고 있으므로, 지속적인 연구 결과물에 대한 모니터링이 필요할 것이다.

## 7. 참고문헌

- [1] The IaaS Framework website, available online at <http://www.iaasframework.com>
- [2] Inocybe Technologies Inc. website, available online at <http://www.inocybe.ca>
- [3] Mathieu Lemay. "Infrastructure as a Service(IaaS), What happened to UCLPv2", APAN workshop, 2008
- [4] Eduard Grasa , Xavier Hesselbach , Sergi Figuerola, et al. "The MANTICORE project: Providing users with a Logical IP Network Service", TNC workshop, 2008



## 8. 부록

본 절에서는 MANTICORE 프로젝트에서 따르는 RFC 4741(NETCONF Configuration Protocol)의 일부를 소개한다.

### ■ RFC 4741의 일부

NETCONF(Network Configuration Protocol)은 네트워크 장치의 설정, 변경, 삭제를 위한 메커니즘을 제공한다. NETCONF는 설정 데이터 및 프로토콜 메시지를 위해 XML-based 데이터 인코딩을 사용한다. NETCONF 오퍼레이션은 RPC(Remote Procedure Call)상에서 구현된다.

NETCONF는 개념적으로 다음 그림과 같은 4 계층으로 구성될 수 있다.

	Layer	Example
(4)	Content	Configuration data
(3)	Operations	<get-config>, <edit-config>
(2)	RPC	<rpc>, <rpc-reply>
(1)	Transport Protocol	BEEP, SSH, SSL, console

위의 계층 중에서 프로토콜 오퍼레이션은 get, get-config, edit-config, copy-config, delete-config, lock, unlock, close-session, kill-session을 포함한다.

get-config는 명시된 설정의 일부 혹은 전부를 검색해 가져오는 온다. 예를 들어, <users> subtree를 가져오기 위해서는 다음과 같은 구문을 사용한다.

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
  <source>
```

```

    <running/>
  </source>
  <filter type="subtree">
    <top xmlns="http://example.com/schema/1.2/config">
      <users/>
    </top>
  </filter>
</get-config>
</rpc>

<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <top xmlns="http://example.com/schema/1.2/config">
      <users>
        <user>
          <name>root</name>
          <type>superuser</type>
          <full-name>Charlie Root</full-name>
          <company-info>
            <dept>1</dept>
            <id>1</id>
          </company-info>
        </user>
        <!-- additional <user> elements appear here... -->
      </users>
    </top>
  </data>
</rpc-reply>

```

edit-config는 명시된 설정의 일부 혹은 전부를 명시된 타겟 설정에 로딩한다. 예를 들어, "Ethernet0/0" 인터페이스에 MTU 크기를 1500으로 설정하기 위해서는 다음과 같은 구문을 사용한다.

```

<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <config>

```

```

    <top xmlns="http://example.com/schema/1.2/config">
      <interface>
        <name>Ethernet0/0</name>
        <mtu>1500</mtu>
      </interface>
    </top>
  </config>
</edit-config>
</rpc>

<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>

```

예를 들어, "Ethernet0/0" 인터페이스의 설정을 제거하기 위해서는 다음과 같은 구문을 사용한다.

```

<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <default-operation>none</default-operation>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <top xmlns="http://example.com/schema/1.2/config">
        <interface xc:operation="delete">
          <name>Ethernet0/0</name>
        </interface>
      </top>
    </config>
  </edit-config>
</rpc>

<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>

```

예를 들어, OSPF area로부터 "192.0.2.4" 인터페이스를 삭제하기 위해서는 다음과 같은 구

문을 사용한다.

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <default-operation>none</default-operation>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
      <top xmlns="http://example.com/schema/1.2/config">
        <protocols>
          <ospf>
            <area>
              <name>0.0.0.0</name>
              <interfaces>
                <interface xc:operation="delete">
                  <name>192.0.2.4</name>
                </interface>
              </interfaces>
            </area>
          </ospf>
        </protocols>
      </top>
    </config>
  </edit-config>
</rpc>

<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```