

ISBN 978-89-6211-473-7 93560

기술매뉴얼

DRAGON SW를 활용한 GMPLS 테스트베드망
상에서의 네트워크 자원할당 시험 보고서

Supercomputing Center
Korea Institute of Science and Technology Information (KISTI)

DRAGON SW를 활용한 GMPLS 테스트베드망 상에서의 네트워크 자원할당 시험 보고서

임 헌 국, 석 우 진, 공 정 욱

한국과학기술정보연구원, 슈퍼컴퓨팅본부, 기반기술개발실
{hklim, wjseok, kju}@kisti.re.kr

최종 수정일 2009년 11월 30일

Abstract

최근 연구망은 의료, 바이오, 항공우주, e-Science 응용과 같은 첨단 응용 연구자들에게 supercomputers, clusters, storage, 가시화 장비 등 네트워크와 연계된 자원들을 네트워크 자원 (대역폭, 경로)과 동시에 동적으로 제공해 줄 수 있는 융합 연구망으로의 변화를 피하고 있다. 그것을 실현해 주기 위해 네트워크 측면에서는 먼저 heterogeneous한 네트워크 환경에서 사용자가 요구시 동적으로 연결 지향형 네트워크를 제공해 줄 수 있어야 하며, 그와 더불어 사용자가 직접 네트워크 자원을 할당 제어해 줄 수 있는 서비스 기술이 제공되어야 한다. Heterogeneous한 네트워크 환경에서 동적으로 연결 지향형 네트워크를 구성하기 위해 GMPLS 제어평면 기술은 가장 보편적인 기술이라 할 수 있다.

본 보고서에서는 GMPLS 테스트베드망 상에서 네트워크 자원의 관리 및 할당을 시험해 보기하기 위해, GMPLS 제어평면을 활용한 서비스 기술인 DRAGON SW를 활용하여 GMPLS 테스트베드 망을 구축 개발하여 보고 구현된 GMPLS 테스트베드망 상에서 네트워크 자원 관리시스템 (NRM)을 통해 네트워크 자원의 동적 할당 및 제어가 가능한지를 시험 테스트 하였다.

Topics

1. 개요
2. DRAGON SW를 활용한 GMPLS 테스트베드망
3. GMPLS 테스트베드망 상에서의 네트워크 자원할당 시험 테스트
4. 결론
5. 참고문헌

1. 개 요

최근 선진 연구망은 의료, 바이오, 항공우주, e-Science 응용과 같은 첨단 응용 연구자들에게 supercomputers, computational clusters, storage array, 가시화 장비 등 네트워크 연계 자원들을 네트워크 자원 (대역폭, 경로)과 동시에 동적으로 제공해 줄 수 있는 융합 연구망으로의 변화를 꾀하고 있다. 그것을 실현해 주기 위해 네트워크 측면에서는 먼저 heterogeneous한 네트워크 환경에서 사용자가 요구시 동적으로 연결 지향형 네트워크를 제공해 줄 수 있어야 하며, 그와 더불어 사용자가 직접 네트워크 자원을 할당 제어해 줄 수 있는 네트워크 자원 할당 및 관리를 위한 서비스 기술이 필수적이라 말할 수 있다. Heterogeneous한 네트워크 환경에서 동적으로 연결 지향형 네트워크를 구성하기 위해 지금까지 소개된 서비스 기술로는 UCLP (User Controlled Lightpath), CHEETHA (Circuit Switched High-Speed End to End Transport Architecture), DRAGON (Dynamic Resource Allocation over GMPLS Optical Networks) 등이 있으며 그중 DRAGON 소프트웨어는 다양한 네트워크 프로토콜이 산재한 이기종 네트워크 환경에서 연결 지향형 네트워크를 꾸며주기 용이한 GMPLS 제어평면에서 동작하도록 설계되었다. 제어평면 아키텍처는 크게 네 부분으로 구성된다. End System Agent 인 CSA, Network Aware Resource Broker인 NARB, Virtual Label Switch Router인 VLSR과 Application Specific Topology Builder 인 ASTB가 그것이다. VLSR은 GMPLS가 지원되지 않는 네트워크 장비가 이해 할 수 있는 관리 언어로 GMPLS 시그널링 메시지를 변환하여 동적으로 VLAN 설정이 가능하도록 해준다.

본 보고서에서는 GMPLS 망 상에서의 네트워크 자원의 관리 및 할당을 테스트하기 위해, 국내 연구망인 KREONET이 GMPLS를 지원하지 않는 환경임을 고려하여 DRAGON 소프트웨어를 활용하여 GMPLS 테스트베드 망을 구축 개발하여 보고 구현된 GMPLS 테스트베드 망 상에서 네트워크 자원관리시스템 (NRM) 시스템을 통해 네트워크 자원의 동적 할당 및 제어가 가능한지를 시험 테스트 하였다. 구현되어진 NRM 시스템은 book-adead 모드 3계층 (GNP, GNR, GRR) 기반의 예약 할당 방식을 적용하였다.

2. DRAGON SW를 활용한 GMPLS 테스트베드망

본 절에서는 DRAGON 소프트웨어의 아키텍처와 메카니즘을 기술한다. DRAGON 소프트웨어는 GMPLS 제어 평면에서 동작하도록 설계되었다. 제어평면 아키텍처는 크게 네 부분으로 구성된다. End System Agent 인 CSA, Network Aware Resource Broker인 NARB, Virtual Label Switch Router인 VLSR과 Application Specific Topology Builder 인 ASTB가 그것이다. 본 장에서는 DRAGON의 구성요소들과 DRAGON SW의 구성과 동작에 대해 간략히 알아보고 Intra-domain 토폴로지 상에서의 각 구성요소의 설정 및 LSP 설정 시험에 대해 기술하기로 한다.

2.1 DRAGON 컴포넌트

2.1.1 CSA

CSA는 데이터 평면의 끝에 있는 end-system에서 실행하는 소프트웨어이다. 이 소프트웨어는 end-to-end 요구기반으로 프로비저닝을 할 경우 GMPLS에 end-system도 직접 참여할 수 있게 지원한다.

2.1.2 VLSR

VLSR는 OSPF-TE, RSVP-TE와 프로비저닝을 시작할 수 있는 사용자 인터페이스를 제공하며, GMPLS를 지원하지 않는 이더넷 스위치들에서 자동 VLAN 설정을 통해 동적인 circuit을 설정할 수 있도록 해준다.

2.1.3 ASTB

ASTB는 응용이 네트워크 서비스를 요구하기 위해 사용할 수 있는 하나의 API를 제공한다. 응용들은 이 API를 DRAGON 네트워크에서 동적인 경로 설정을 요구하는데 사용할 수 있다. 현재 이 API는 IPv4 주소로 표현되는 출발지와 목적지 사이에 전용 네트워크 경로를 설정하도록 요구할 수 있는 간단한 메시지의 집합이다. 차후 토폴로지 빌더를 포함할 경우 특정 응용만의 전용 네트워크 토폴로지를 여러 개의 LSP로 구현이 가능하다.

2.1.4 NARB

NARB의 핵심 기능은 도메인간 라우팅, 도메인간 경로 계산, 토폴로지 추상화와 같은 고급 기능들이다. NARB의 구성요소인 RCE (Resource Computation Element)는 경로 계산의 기능을 수행한다. NARB는 도메인간 라우팅 정보와 LSP 프로비저닝을 위한 토폴로지 정보를 교환한다. 즉, 자신의 네트워크에 대한 OSPF와 TE를 포함한 정보를 GMPLS 네트워크 상에 있는 인접한 이웃 NARB 과 교환함으로써 서로의 링크 상태 정보 데이터베이스를 공유한다.

2.2 DRAGON SW의 구성과 동작

이상에서 살펴 본 바와 같이 DRAGON은 여러 개의 소프트웨어 패키지들의 기능을 통합적으로 활용하여 원하는 기능을 수행하도록 한다. Figure 1은 지금까지 살펴본 구성요소들의 기능과 인터페이스를 하나의 그림으로 표현한 것이다. 본 절에서는 2.1절에서 기술한 각 구성요소가 하나의 LSP를 설정하기 위해 어떻게 동작하는 지 전체적인 동작 과정을 설명한다.

동일한 소프트웨어 패키지로 구성되는 CSA와 VLSR은 CSA가 Peer-to-Peer 모드 즉 라우팅에 참여할 경우 dragon, zebra, ospfd, RSVPD 등 네 개의 데몬 프로세스를 실행한다. ASTB와 통신할 일이 있을 경우 여기에 node_agent 데몬이 추가로 실행된다. zebra는 Linux 라우팅 소프트웨어 패키지의 프로세스로서, 네트워크 인터페이스와 라우팅 테이블을 관리한다. ospfd는 shared memory로 zebra와 통신함으로써, 라우팅 정보를 갱신한다. dragon 데몬은 DRAGON에 특화된 vty를 제공하는 서비스 데몬으로 사용자의 요구를 받아 이를 RSVP와 유사한 API로 변환한 다음 RSVPD로 전달한다. 만일 도메인간 라우팅일 경우, NARB API로 직접 NARB 서버와 통신하여 ERO를 가져온다.

인트라 도메인 라우팅일 경우 설정에 따라 내부의 ospfd에 요청하여 ERO를 받아 오기도 하고 NARB 서버에 ERO를 직접 요청하기도 한다. CSA가 Proxy 모드일 경우 CSA는 라우

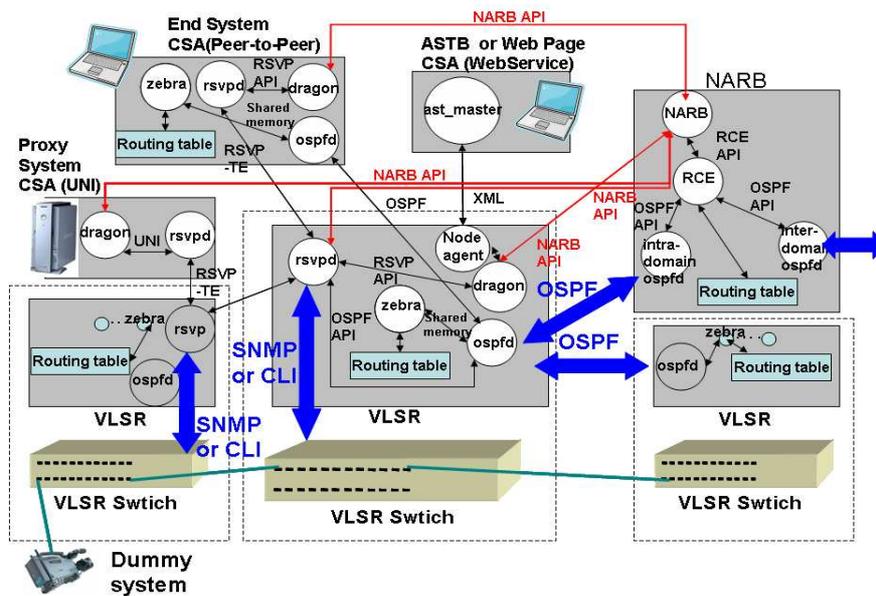


Figure 1. DRAGON 소프트웨어 인터페이스

팅에 직접 참여하지 않으므로, dragon과 rsvpd 만 실행한다. RSVPD가 PATH 메시지를 보낼 next hop은 dragon 데몬이 직접 NARB API 로 NARB 서버에 요청하여 받은 ERO를 통해 결정된다.

NARB 시스템의 intra-domain ospfd는 border VLSR의 ospfd와 LSA를 주고받음으로써, 도메인 내부의 네트워크 토폴로지 정보를 얻는다. inter-domain ospfd는 이웃 NARB의 intra-domain ospfd와 LSA를 주고받음으로써, 도메인 간 네트워크 토폴로지 정보를 얻는다.

Figure 2은 각 데몬 프로세스들과 데몬 프로세스들 사이에 데이터 흐름을 보여준다. 데몬 서버들이 사용하는 포트는 dragon 2611, ospfd 2604, narb 2626, RCE 2688 이다.

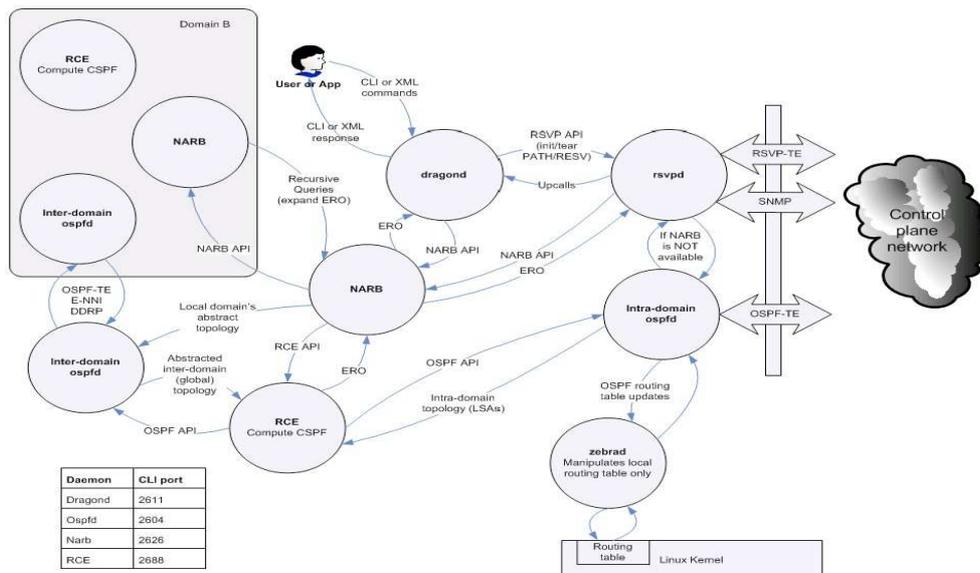


Figure 2. Dragon 소프트웨어 데이터 흐름

2.3 Intra-domain 토폴로지 상에서의 DRAGON SW 설정 및 시험

본 절에서는 cisco3750 간의 동종 이더넷 스위치 장비에 대해, 하나의 NARB를 가진 도메인 내에서의 LSP 설정 시험에 대해 기술하기로 한다. Figure 3은 이를 위한 네트워크 토폴로지를 보여준다. 기본 설정에서와 같이 C3500 스위치는 스위칭 허브로 사용되었고, 두 이더넷 스위치는 Gi1/0/11 과 Gi1/0/1 포트로 서로 연결되어 있다. 전체 시스템들은 NARB와 모두 연결되어 있다. 그러나, VLSR1만이 GRE 터널을 통해 OSPF-TE 통신을 수행한다. 하나의 NARB로 관리하고 있는 네트워크 내의 모든 VLSR이 NARB와 OSPF-TE를 수행할 필요는 없다. 이 VLSR을 border VLSR이라 하며, 두 대의 종단 시스템은 UNI 모드 CSA가 아니라 Peer 모드 CSA로 동작하도록 설정하기로 하였다.

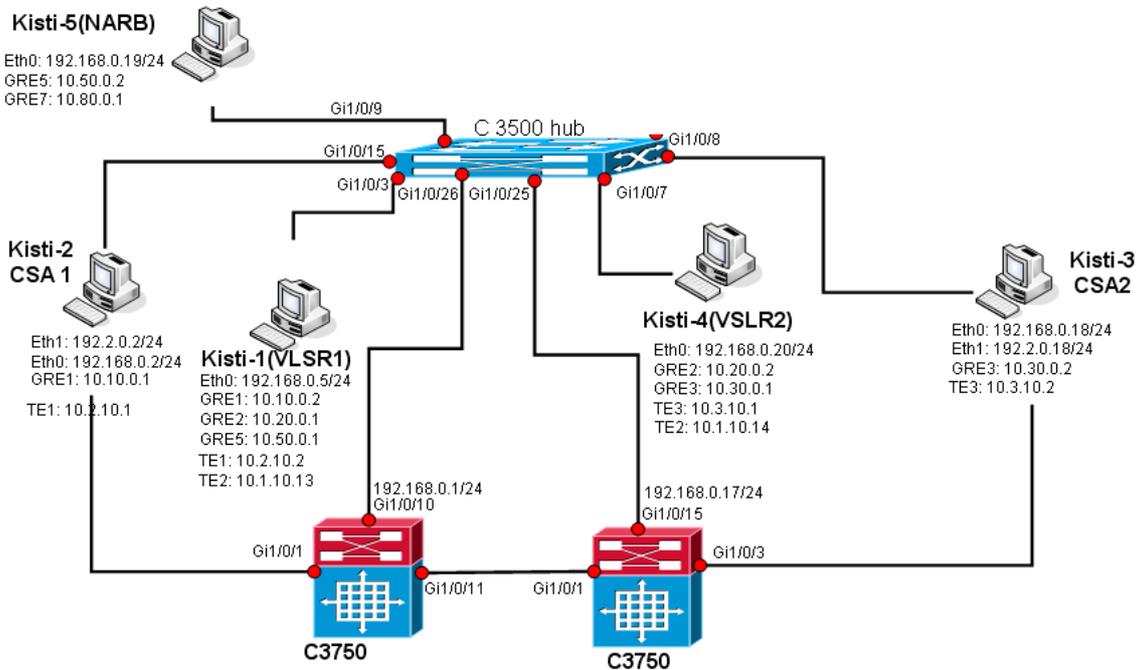


Figure 3. Intra-domain Test Network Topology

2.3.1 GRE 설정

GRE1은 CSA1과 VLSR1의 제어평면 통신을 위해, GRE2는 VLSR1과 VLSR2, GRE3는 VLSR2와 CSA2, GRE5는 border VLSR인 VLSR1과 NARB1의 제어평면 통신을 위해 필요하다. 각각의 장비에서는 이들 GRE를 설정해 주어야 한다. Figure 3에서 보여지는 바와 같이 각 Dragon component 간에 OSPF-TE 정보와 RSVP 시그널링 정보를 서로 주고받기 위한 GRE 터널 설정이 필요하며, 총 4개의 GRE 터널이 필요하였다.

2.3.2 소프트웨어 설정 파일

2.3.2.1 CSA1 설정 파일

CSA1에서 동작을 위해 설정해 주어야 하는 파일은 dragon.conf, ospfd.conf,

RSVPD.conf, zebra.conf 로 모두 4개이다. 다음은 각각의 설정 파일에 대한 설정이다.

1) dragon.conf

Peer to peer CSA 모델로 dragon.conf 파일은 DRAGON 소프트웨어 실행 시 dragon 데몬이 참조하는 파일이다.

```
! -- dragon --
! DRAGON sample configuration file
hostname csal-dragon
password dragon
```

2) ospfd.conf

```
! -- ospf --
! OSPFd sample configuration file
hostname csa-ospf
password dragon
enable password dragon
log stdout
! -- sample ospf configuration for a dragon VLSR controlling a Layer2 Ethernet switch --
!
interface gre1
description GRE tunnel between csal and vlsr1
ip ospf network point-to-point
!
router ospf
ospf router-id 192.168.0.2
network 10.10.0.0/30 area 0.0.0.0
ospf-te router-address 192.168.0.2
ospf-te interface gre1
    level gmpls
    data-interface ip 10.2.10.1
    swcap l2sc encoding ethernet
    max-bw 125000000
    max-rsv-bw 125000000
    max-lsp-bw 0 125000000
    max-lsp-bw 1 125000000
    max-lsp-bw 2 125000000
    max-lsp-bw 3 125000000
    max-lsp-bw 4 125000000
    max-lsp-bw 5 125000000
    max-lsp-bw 6 125000000
    max-lsp-bw 7 125000000
    vlan 100 to 200
    metric 10
exit
```

```
!  
line vty  
!
```

3) RSVPD.conf

RSVPD 데몬 프로세스가 기동 시 참조하는 파일로 시그널링을 위한 GRE 인터페이스를 설정할 수 있다.

```
interface gre1 tc none mpls  
api 4000
```

4) zebra.conf

```
! *- zebra *-  
! zebra sample configuration file  
! $Id: zebra.conf.sample,v 1.14 1999/02/19 17:26:38 developer Exp $  
!  
hostname csal-zebra  
password zebra  
enable password zebra  
!  
interface lo  
interface gre1  
!  
line vty  
log file /var/log/zebra.log
```

2.3.2.2 CSA2 설정 파일

CSA2도 CSA1과 마찬가지로 Peer to peer 모델로 동작하며, 이를 위해 설정해 주어야 하는 파일은 dragon.conf, ospfd.conf, RSVPD.conf, zebra.conf 로 모두 4개이다.

1) dragon.conf

```
! *- dragon *-  
! DRAGON sample configuration file  
!  
hostname csa2-dragon  
password dragon
```

2) ospfd.conf

Peer-to-peer CSA 모드로 작동하는 CSA2은 라우팅에 참여하므로, VLSR 과 마찬가지로 ospfd.conf 의 설정이 필요하다.

```

! -- ospf --
!
! OSPFd sample configuration file
!
hostname vlsr-ospf
password dragon
enable password dragon
log stdout
log file /var/log/ospfd.log
!
! -- sample ospf configuration for a dragon VLSR controlling a Layer2 Ethernet switch
--
!
interface gre3
description GRE tunnel between csa2(kisti-3) and vlsr2(kisti-4)

ip ospf network point-to-point
!
router ospf
ospf router-id 192.168.0.18
network 10.30.0.0/30 area 0.0.0.0
ospf-te router-address 192.168.0.18
ospf-te interface gre3
    level gmpls
    data-interface ip 10.3.10.2
    swcap l2sc encoding ethernet
    max-bw 125000000
    max-rsv-bw 125000000
    max-lsp-bw 0 125000000
    max-lsp-bw 1 125000000
    max-lsp-bw 2 125000000
    max-lsp-bw 3 125000000
    max-lsp-bw 4 125000000
    max-lsp-bw 5 125000000
    max-lsp-bw 6 125000000
    max-lsp-bw 7 125000000
    vlan 100 to 200
    metric 10
    exit
!
line vty
!

```

3) RSVPD.conf

```
interface gre3 tc none mpls
api 4000
```

4) zebra.conf

```
! -- zebra --
! zebra sample configuration file
! $Id: zebra.conf.sample,v 1.14 1999/02/19 17:26:38 developer Exp $
!
hostname csa2-zebra
password zebra
enable password zebra
! Interface's description
interface lo
interface gre3
!
line vty
log file /var/log/zebra.log
```

2.3.2.3 VLSR1 설정 파일

VLSR1도 CSA1과 마찬가지로 동작을 위해 설정해 주어야 하는 파일은 dragon.conf, ospfd.conf, RSVPD.conf, zebra.conf 로 모두 4개이다. 그러나, 설정의 내용은 CSA와 다르다. VLSR은 자신이 제어하고 있는 스위치의 포트 정보에 대해 알고 있어야 한다. 다음은 각각의 설정파일의 내용이다.

1) dragon.conf

dragon.conf의 내용은 CSA1이나 CSA2와 다르지 않다. prompt 에 vlsr 임을 명시하기 위해 hostname을 vlsr1로 설정하였다. 또한 NARB 와의 통신을 위해 NARB 의 IP address 를 명시해 주어야 한다.

```
! -- dragon --
!
! DRAGON sample configuration file
!
hostname vlsr1-dragon
password dragon
configure narb intra-domain ip-address 192.168.0.19 port 2609
```

2) ospfd.conf

VLSR1의 ospfd.conf는 CSA의 그것과 다르다. 먼저 CSA1, VLSR2, NARB와의 인터페이스 GRE를 모두 명시해야 한다. 또한 각 ospf-te interface gre1, interface gre2의

data-interface 설정에 실제 스위치의 ip 와 port 정보가 포함되어 있다. 이 중 switch-port 는 정수로만 입력할 수 있다. 각 스위치 마다 이 정수를 해석하는 방법은 다르지만, cisco 3750 스위치의 경우는 슬롯이 1개이고 포트 숫자가 곧 정수이므로 별도의 포트 변환 규칙은 필요하지 않다.

```
! -- ospf --
!
! OSPFd sample configuration file
!
hostname vlsr1-ospf
password dragon
enable password dragon
log stdout
log file /var/log/ospfd.log
!
! NOTE: max. bandwidth parameters are in bytes/sec, for example:
! 1 Gbps = 125000000 bytes/sec
! 10 Gbps = 1250000000 bytes/sec
!
! -- sample ospf configuration for a dragon VLSR controlling a Layer2 Ethernet switch --
!
interface gre1
description GRE tunnel between vlsr1(kisti-1) and csal(kisti-2)
ip ospf network point-to-point
!
interface gre2
description GRE tunnel between vlsr1(kisti-1) and vlsr2(kisti-4)
ip ospf network point-to-point
!
interface gre5
description GRE tunnel between narb-server(Kisti-5) and vlsr1(kisti-1)
ip ospf network point-to-point
!
router ospf
ospf router-id 192.168.0.5
network 10.10.0.0/30 area 0.0.0.0
network 10.20.0.0/30 area 0.0.0.0
network 10.50.0.0/30 area 0.0.0.0
  ospf-te router-address 192.168.0.5
  ospf-te interface gre1
    level gmpls
    data-interface ip 10.2.10.2 protocol snmp switch-ip 192.168.0.1 switch-port 1
    swcap l2sc encoding ethernet
    max-bw 125000000
    max-rsv-bw 125000000
```

```

max-lsp-bw 0 125000000
max-lsp-bw 1 125000000
max-lsp-bw 2 125000000
max-lsp-bw 3 125000000
max-lsp-bw 4 125000000
max-lsp-bw 5 125000000
max-lsp-bw 6 125000000
max-lsp-bw 7 125000000
vlan 100
vlan 200
metric 10
exit
!
ospf-te interface gre2
level gmpls
data-interface ip 10.1.10.13 protocol snmp switch-ip 192.168.0.1 switch-port 11
swcap l2sc encoding ethernet
max-bw 125000000
max-rsv-bw 125000000
max-lsp-bw 0 125000000
max-lsp-bw 1 125000000
max-lsp-bw 2 125000000
max-lsp-bw 3 125000000
max-lsp-bw 4 125000000
max-lsp-bw 5 125000000
max-lsp-bw 6 125000000
max-lsp-bw 7 125000000
vlan 100
vlan 200
metric 10
exit
!
line vty
!
```

3) RSVPD.conf

CSA1, VLRS2, NARB 와의 시그널링 인터페이스를 모두 기술해 준다.

```

interface gre1 tc none mpls
interface gre2 tc none mpls
interface gre5 tc none mpls
api 4000
```

4) zebra.conf

존재하는 모든 GRE 인터페이스를 정의한다.

```

! -- zebra --
! zebra sample configuration file
hostname vlsr1-zebra
password zebra
enable password zebra
!
! Interface's description.
!
interface lo
interface gre1
interface gre2
interface gre5
!
line vty
log file /var/log/zebra.log

```

2.3.2.4 VLSR2 설정 파일

1) dragon.conf

VLSR2는 border VLSR 이 아니므로 VLSR1과 같은 NARB 정보가 필요없다. 그러나, VLSR1과 마찬가지로 동작을 위해 설정해 주어야 하는 파일은 dragon.conf, ospfd.conf, RSVPD.conf, zebra.conf 로 모두 4개가 필요하다.

```

! -- dragon --
! DRAGON sample configuration file
!
hostname kisti4-dragon
password dragon

```

2) ospfd.conf

VLSR2가 제어하는 장비는 cisco 3750 스위치이다. 따라서, VLSR1과 마찬가지로 VLSR2가 연결된 포트와 데이터 인터페이스를 명시해야 한다. 먼저 VLSR1, CSA2와의 인터페이스 GRE를 모두 명시해야 한다. 또한 각 ospf-te interface gre1, interface gre2의 data-interface 설정에 실제 스위치의 ip 와 port 정보가 포함되어 있다.

```

! -- ospf --
! OSPFd sample configuration file
!
hostname vlsr-ospf
password dragon
enable password dragon
log stdout
log file /var/log/ospfd.log
!
! NOTE: max. bandwidth parameters are in bytes/sec, for example:

```

```

! 1 Gbps = 125000000 bytes/sec
! 10 Gbps = 1250000000 bytes/sec
! *- sample ospf configuration for a dragon VLSR controlling a Layer2 Ethernet switch *-

!
interface gre2
description GRE tunnel between vlsr1(kisti-1) and vlsr2(kisti-4)
ip ospf network point-to-point
!
interface gre3
description GRE tunnel between csa2(kisti-3) and vlsr2(kisti-4)
ip ospf network point-to-point
!
router ospf
ospf router-id 192.168.0.20
network 10.20.0.0/30 area 0.0.0.0
network 10.30.0.0/30 area 0.0.0.0
  ospf-te router-address 192.168.0.20
  ospf-te interface gre2
    level gmpls
    data-interface ip 10.1.10.14 protocol snmp switch-ip 192.168.0.17 switch-port 1
    swcap l2sc encoding ethernet
    max-bw 125000000
    max-rsv-bw 125000000
    max-lsp-bw 0 125000000
    max-lsp-bw 1 125000000
    max-lsp-bw 2 125000000
    max-lsp-bw 3 125000000
    max-lsp-bw 4 125000000
    max-lsp-bw 5 125000000
    max-lsp-bw 6 125000000
    max-lsp-bw 7 125000000
    vlan 100
    vlan 200
    metric 10
  exit
!
ospf-te interface gre3
  level gmpls
  data-interface ip 10.3.10.1 protocol snmp switch-ip 192.168.0.17 switch-port 3
  swcap l2sc encoding ethernet
  max-bw 125000000
  max-rsv-bw 125000000
  max-lsp-bw 0 125000000
  max-lsp-bw 1 125000000
  max-lsp-bw 2 125000000

```

```

max-lsp-bw 3 125000000
max-lsp-bw 4 125000000
max-lsp-bw 5 125000000
max-lsp-bw 6 125000000
max-lsp-bw 7 125000000
vlan 100
vlan 200
metric 10
exit
!
line vty
!
```

2) RSVPD.conf

시그널링 인터페이스를 모두 기술해 준다.

```

interface gre2 tc none mpls
interface gre3 tc none mpls
api 4000
```

3) zebra.conf

```

! -- zebra --
! zebra sample configuration file
!
! $Id: zebra.conf.sample,v 1.14 1999/02/19 17:26:38 developer Exp $
!
hostname kisti4-vlsr2
password zebra
enable password zebra
!
! Interface's description.
!
interface lo
interface gre2
interface gre3
!
line vty
log file /var/log/zebra.log
```

2.3.2.5 NARB1 설정 파일

NARB를 기동하기 위해서는 narb.conf, ospfd-intra.conf, ospfd-inter.conf, zebra.conf, rce.conf가 필요하다. 또한 rce 데몬 프로세스의 정상적인 동작을 위해

schema_combo.rsd.sample 을 schema_combo.rsd로 복사해 둔다.

1) narb.conf

narb.conf 는 narb 데몬 프로세스가 기동 시 참조하는 파일이다.

domain-id 는 그 도메인을 표현한다. Figure 3과 같이 현재 NARB가 관리하고 있는 도메인은 192.168.0.0 도메인이다. 현재 NARB은 하나의 도메인만 관장할 뿐 다른 어떤 NARB와의 연결도 가지지 않는다. 따라서, intra-domain-ospfd 에 관한 정보만 명시한다. originate-interface 는 VLSR1의 ospfd 데몬과 네트워크 토폴로지 정보를 교환하기 위해 사용하고 있는 GRE 인터페이스의 IP 주소를 의미한다.

```
!  
domain-id {ip 192.168.0.0}  
!  
intra-domain-ospfd {address localhost port 2617  
    originate-interface 10.50.0.2 area 0.0.0.0}  
!  
router {id 192.168.0.5  
    inter-domain link to vlsr2  
    link {id 192.168.0.20 type 1  
        max_bw 1250.0 max_rsv_bw 1250.0  
        unrsv_bw0 1250.0 unrsv_bw1 1250.0 unrsv_bw2 1250.0 unrsv_bw3 1250.0  
unrsv_bw4 1250.0 unrsv_bw5 1250.0 unrsv_bw6 1250.0 unrsv_bw7 1250.0  
        enc_type 2 sw_type 51  
        metric 10  
        local_if 10.1.10.13 remote_if 10.1.10.14  
        vlan_tags(100:100)  
    }  
}  
!  
cli {host Kisti5-narb password dragon}  
!
```

2) ospfd-intra.conf

ospfd-intra.conf는 도메인 내부의 네트워크 토폴로지 정보를 교환하기 위해 VLSR1의 ospfd 데몬 프로세스와 통신하는 NARB의 ospfd 데몬 프로세스가 기동 시 참조하는 파일이다.

```
! Intra-domain ospfd configuration for narb  
! 2005/03/11 16:32:10  
!  
hostname Kisti5-ospf-intra  
password dragon  
log stdout  
!
```

```

interface gre5
  description GRE tunnel between Kisti5-narb and kisti1-vlsr
  ip ospf network point-to-point
!
router ospf
  ospf router-id 192.168.0.19
  network 10.50.0.0/30 area 0.0.0.0
    ospf-te router-address 192.168.0.19
!
line vty

```

3) ospfd-inter.conf

다른 NARB와의 도메인 간 연결이 없으므로 필요한 파일은 아니다. 그러나 NARB 시스템 기동시 inter-domain을 위한 ospfd 데몬 프로세스가 실행되므로, 기본적인 호스트 이름과 패스워드만 포함한 파일로 존재해야 한다.

```

hostname Kisti5-ospf-inter
password dragon
log stdout
!

```

4) zebra.conf

ospfd 데몬 프로세스에게 인터페이스 정보를 주고, 라우팅 정보를 업데이트 하기 위해 기동하는 zebra 데몬 프로세스가 기동 시 참조하는 파일이다.

```

! -- zebra --
!
! zebra sample configuration file
!
! $Id: zebra.conf.sample,v 1.14 1999/02/19 17:26:38 developer Exp $
!
hostname Kisti5-zebra
password zebra
enable password zebra
!
! Interface's description.
!
interface lo
interface gre5
!
line vty
log file /var/log/zebra.log

```

5) rce.conf

Resource Computation Engine 인 rce 데몬 프로세스가 기동 시 참조하는 파일이다. 도메인 아이디와 트래픽 엔지니어링 테이블을 위한 스키마 파일에 대한 설정이 필요하다.

```
domain-id {ip 192.168.0.0}
!  
include-tedb-schema {path/usr/local/dragon/etc/schema_combo.rsd}
!
```

2.3.3 네트워크 스위치 설정

사용하고자 하는 VLAN id를 cisco 3750의 데이터베이스에 등록해 둔다. 그러나 cisco 3750의 경우, 이러한 데이터베이스 등록을 따로 하지 않더라도, 랜덤으로 vlan id를 생성하기 때문에 LSP 설정에 문제가 없다.

2.3.4 DRAGON 소프트웨어 기동과 종료

```
[root@kistil bin]#./dragon.sh start-vlsr
```

명령을 실행하고 나면, 제대로 기동하였는지 확인하여야 한다. 리눅스 명령어를 사용하면, 다음과 같이 모두 4개의 프로세스가 떠 있음을 확인할 수 있다.

```
[root@kistil bin]# ps -ef | grep dragon  
root      6448  5778  0 15:29 pts/4    00:00:00 grep dragon  
root      6443      1  0 15:28 ?          00:00:00 /usr/local/dragon/sbin/zebra -d -f  
/usr/local/dragon/etc/zebra.conf  
root      6441      1  0 15:28 ?          00:00:00 /usr/local/dragon/bin/RSPVD -c  
/usr/local/dragon/etc/RSPVD.conf -d -o /var/log/RSPVD.log -L select,ref,packet  
root      6444      1  0 15:28 ?          00:00:00 /usr/local/dragon/bin/dragon -d -f  
/usr/local/dragon/etc/dragon.conf  
root      6445      1  0 15:28 ?          00:00:00 /usr/local/dragon/sbin/ospfd -d -f  
/usr/local/dragon/etc/ospfd.conf
```

VLSR1과 VLSR2는 dragon.sh 의 start-vlsr 옵션으로 기동하면 된다.

NARB은 다음의 명령으로 실행가능하다.

```
[root@kisti-5 bin]# ./dragon.sh start-narb
```

프로세스들의 정상적인 기동을 알아보기 위해 ps 명령어를 실행하면 된다.

```
[root@kisti-5 bin]# ps -ef | grep dragon  
root      13185      1  0 15:43 ?          00:00:00 /usr/local/dragon/sbin/zebra -d -f
```

```

/usr/local/dragon/etc/zebra.conf
root    13187    1  0 15:43 ?          00:00:00 /usr/local/dragon/sbin/ospfd -d -I -P
2614 -f /usr/local/dragon/etc/ospfd-inter.conf
root    13189    1  0 15:43 ?          00:00:00 /usr/local/dragon/sbin/ospfd -d -P 2604
-f /usr/local/dragon/etc/ospfd-intra.conf
root    13203    1  0 15:43 ?          00:00:00 /usr/local/dragon/sbin/rce -d -f
/usr/local/dragon/etc/rce.conf
root    13207    1  0 15:43 ?          00:00:00 /usr/local/dragon/sbin/narb -d -f
/usr/local/dragon/etc/narb.conf
root    13209 12398  0 15:43 pts/0    00:00:00 grep dragon

```

NARB는 narb, rce, ospfd 의 라우팅 정보를 갱신하고 인터페이스 정보를 전달하기 위한 zebra와 두 개의 ospfd(intra-domain, inter-domain) 모두 5개의 데몬 프로세스를 실행하고 있다.

2.3.5 LSP의 설정

모든 설정이 끝나고, DRAGON 소프트웨어의 기동이 끝나면, LSP를 설정한다. LSP의 설정은 dragon vty 에 접속하여 수행한다. LSP 설정 전 CSA1과 CSA2 두 시스템은 데이터 평면의 연결이 없으므로, ping은 실패하게 되며, LSP 설정을 위해 csa1의 dragon 데몬 vty에 접속한다.

```

[root@kisti-2 ~]# telnet 192.168.0.2 2611
Trying 192.168.0.2...
Connected to 192.168.0.2.
Escape character is '^]'.

```

```

_____
|  _ W _ _ _ _ _ _ _ _ _ _
| | | | ' _ / _ ` | / _ W | ' W
| | | | | ( | | ( | | ( | | | |
| ___/|_| W _ _ | W _ _ | W ___/|_| |_|
          |___/

(D)ynamic (R)esource (A)llocation via
(G)MPLS (O)ptical (N)etworks
Based on Zebra (version 0.94).
Copyright 1996-2002 Kunihiro Ishiguro.
Copyright 2003-2004 the Dragon Team.

```

User Access Verification

```

Password:
csa1-dragon>

```

성공적으로 vty 에 접속되면, LSP test을 만든다. peer-to peer 모드일 때, LSP 의 설정을 테스트하기 위해 lsp-id 와 tunnel-id를 설정하여야 한다.

```

csal-dragon> edit lsp test
csal-dragon(edit-lsp-test)# set source ip-address 192.168.0.2 lsp-id 1000 destination
ip-address 192.168.0.18 tunnel-id 2000
csal-dragon(edit-lsp-test)# set bandwidth gige swcap l2sc encoding ethernet gpid ethernet
csal-dragon(edit-lsp-test)# set vtag 100
csal-dragon(edit-lsp-test)# exit
csal-dragon> commit lsp test
csal-dragon> show lsp

```

LSP status summary

Name	Status	Dir	Source (IP/LSP ID)	Destination (IP/Tunnel ID)
test	In service	<=>	192.168.0.2 1000	192.168.0.18 2000

csal-dragon>

show lsp는 현재 존재하는 lsp의 내용을 요약해서 보여준다. LSP가 제대로 생성되었는지 확인하기 위해서는 현재의 프롬프트 상에서 LSP test 는 in-service 상태여야 한다. 제대로 수행이 되었다면 각 스위치에서 vlan이 제대로 설정되었는지 확인한다.

2) cisco3750 #1

KISTI_TEST#show vlan

VLAN Name	Status	Ports
2 VLAN	active	Gi 1/0/1, Gi 1/0/11
1002 fddi-default	act/unsup	
1003 token-ring-default	act/unsup	
1004 fddinet-default	act/unsup	
1005 trnet-default	act/unsup	

VLAN Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
2 enet	100001	1500	-	-	-	-	0	0	
1002 fddi	101002	1500	-	-	-	-	0	0	
1003 tr	101003	1500	-	-	-	-	0	0	
1004 fdnet	101004	1500	-	-	-	ieee	0	0	
1005 trnet	101005	1500	-	-	-	ibm	0	0	

2) cisco3750 #2

Switch#show vlan

VLAN Name	Status	Ports
100 VLAN	active	Gi 1/0/1, Gi 1/0/3
1002 fddi-default	act/unsup	
1003 token-ring-default	act/unsup	
1004 fddinet-default	act/unsup	
1005 trnet-default	act/unsup	

VLAN Type	SAID	MTU	Parent	RingNo	BridgeNo	Stp	BrdgMode	Trans1	Trans2
100 enet	100001	1500	-	-	-	-	0	0	
1002 fddi	101002	1500	-	-	-	-	0	0	
1003 tr	101003	1500	-	-	-	-	0	0	
1004 fdnet	101004	1500	-	-	-	ieee	0	0	
1005 trnet	101005	1500	-	-	-	ibm	0	0	

마지막으로 이러한 설정이 제대로 되었는지 중단 시스템에서 ping으로 확인한다.

```
[root@kisti-2 ~]# ping 192.2.0.18
PING 192.2.0.18 (192.2.0.18) 56(84) bytes of data.
64 bytes from 192.2.0.18: icmp_seq=1 ttl=64 time=2.46 ms
64 bytes from 192.2.0.18: icmp_seq=2 ttl=64 time=0.216 ms
64 bytes from 192.2.0.18: icmp_seq=3 ttl=64 time=1.36 ms
64 bytes from 192.2.0.18: icmp_seq=4 ttl=64 time=0.219 ms
64 bytes from 192.2.0.18: icmp_seq=5 ttl=64 time=1.36 ms
64 bytes from 192.2.0.18: icmp_seq=6 ttl=64 time=1.32 ms
```

3. GMPLS 테스트베드망 상에서의 네트워크 자원할당 시험 테스트

본장에서는 네트워크 자원 관리 시스템을 통해 DRAGON SW를 활용하여 구축한 GMPLS 테스트베드망 상에서의 네트워크 자원의 할당 제어를 시험해 보기로 한다. 네트워크 자원 할당 제어 시험을 위해 위탁과제를 통해 개발되어진 네트워크 자원관리시스템 (NRM)을 활용하였다.

3.1 NRM Architecture

Figure 4는 그리드 네트워크 경로와 자원의 생성 및 예약과 할당을 수행하는 NRM (Network Resource Manager) 시스템의 구성도와 외부 인터페이스를 나타낸다.

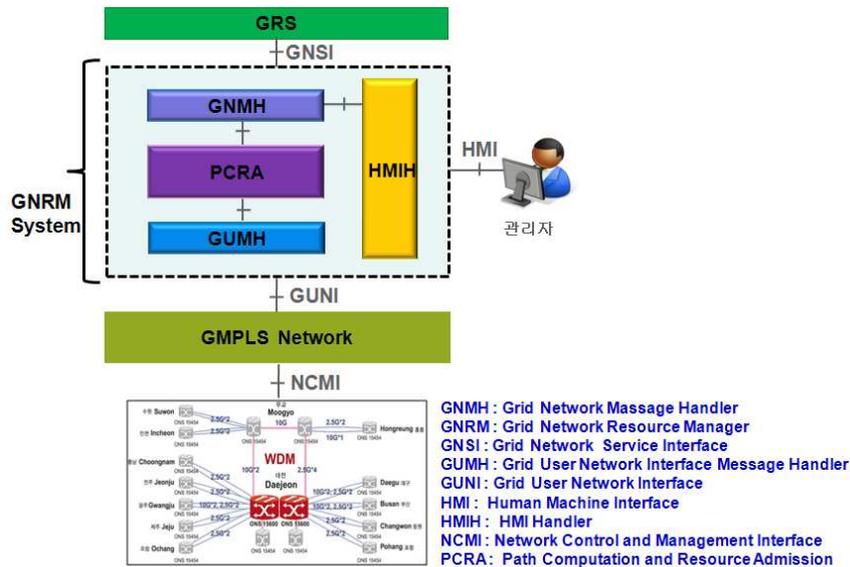


Figure 4. NRM System Configuration and External Interfaces

NRM 시스템은 GNMH (Network Message Handler), PCRA(Path Computation and Resource Admission), GUMH(G-UNI Message Handler) 그리고 HMIH(Human Machine Interface Handler) 블록으로 구성된다. 외부 인터페이스로 WSRF 기반의 GNSI(Grid Network Service Interface), 웹 페이지 기반의 HMI(Human Machine Interface) 그리고 특정 전달 망이나 장비의 제어 기술에 독립적으로 정의된 GUNI(Grid User Network Interface)가 있다. 글로벌 자원 스케줄러(GRS)와는 단일 또는 두 단계 자원 예약 메커니즘을 이용하는 GNSI 인터페이스가 사용된다. 관리자와의 인터페이스인 HMI는 기본적으로 GNSI의 모든 기능을 수용하며, 부가적으로 정책의 설정 및 해제와 GMPLS 망에 대한 관리 기능이 정의된다. GUNI는 예약된 그리드 네트워크의 자원의 할당 및 해제를 위하여 GMPLS 제어평면과의 인터페이스를 수행한다.

GNMH 블록은 GNSI 인터페이스와 HMI 인터페이스 요청을 분석하여 PCRA로 전달하는 기능을 수행한다. PCRA는 네트워크 자원의 예약을 위하여 경로 계산과 자원의 수락 기능을 수행한다. 추상화된 토폴로지를 구성하기 위하여 네트워크 경로를 생성하거나 삭제하고 네트워크의 경로 위에서 네트워크 자원의 생성과 해제 기능을 수행한다. 또한, 생성된 네트워크 자원에 대하여 시간대 별로 예약을 수행하며 예약 된 자원의 할당과 해제를 위하여 GUMH 블록과 인터페이스를 수행한다. GUMH 블록은 GUNI 인터페이스를 통하여 GMPLS 망과 LSP의 설정 및 해제를 위한 제어 정보를 교환한다. HMIH 블록은 HMI 인터페이스를 통하여 관리자가 요청하는 네트워크 경로와 자원의 관리 및 예약과 정책 설정을 위하여 컴맨드 라인 및 웹 기반 사용자 인터페이스 기능을 제공한다. 또한, GUNI 인터페이스를 이용하여 GMPLS 제어망의 토폴로지와 LSP 관리 기능을 제공한다.

개발된 NRM 시스템은 e-Science와 같은 융합망 어플리케이션을 위해 book-adead 모드 3계층 (GNP, GNR, GRR) 기반의 예약 할당 방식을 적용하였으며 이는 다른 선진국의 융합망 프로젝트에서 개발되어진 NRM 시스템에서 적용한 2계층 방식 (GNP, GRR)의 예약 할당 방식과 차별화된 것이다.

3.2 GMPLS 테스트베드망 상에서의 네트워크자원관리시스템 연동 구성도

Figure 5는 NRM과 GMPLS 테스트베드망과의 연동 구성도이며, 2장에서 기술한 Intradomain 토폴로지 내에서의 GMPLS 테스트베드망과 개발된 NRM 시스템을 연동한 것이다.

CSA는 제어 평면의 중단 사이트 노드에서 LSP의 설정과 해제를 수행한다. 기존 망에서 GMPLS 제어 기능이 없는 스위치나 라우터 장비를 위하여 VLSR은 GMPLS의 프록시로 동작한다. 2장에서 기술한대로 CSA, VLSR 및 NARB 사이의 시그널링을 위하여 Figure 5의 GRE 터널을 사용한다. GRE 터널을 통해 RSVP-TE reservation 메시지와 ospf-te 라우팅 메시지를 주고 받도록 DRAGON SW 패키지를 활용하여 상부 제어평면을 구성하도록 하였다. 하부 데이터평면을 위한 전달망은 가상의 LAN(VLAN)이 지원되는 시스코의 3750 스위

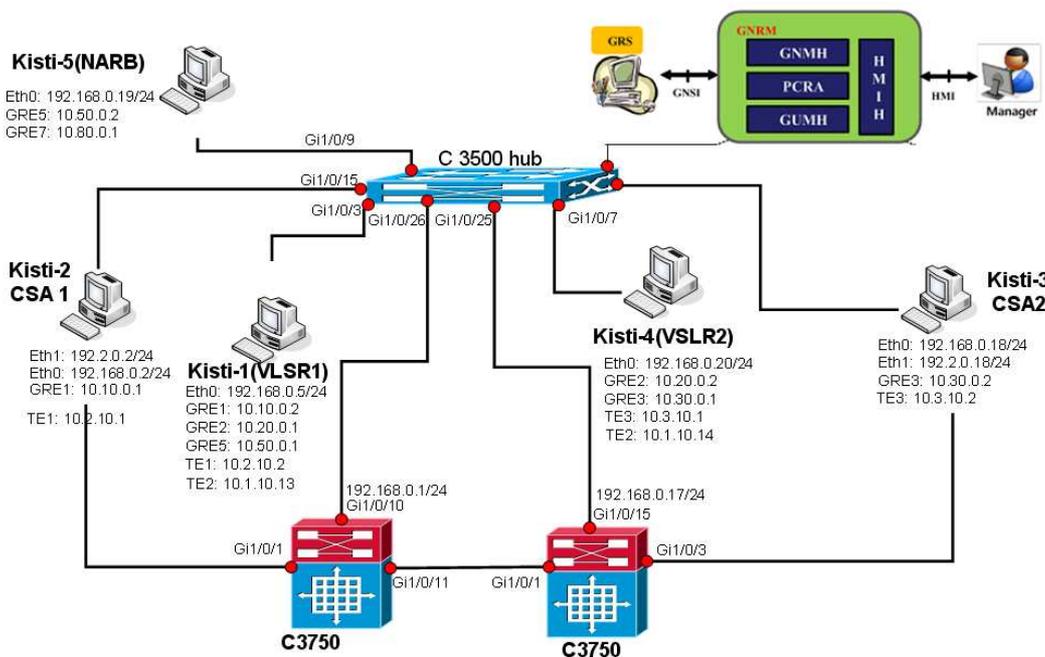


Figure 5. GMPLS 테스트베드망 상에서의 네트워크자원관리시스템 구성도

치를 사용하여 구성하였다.

NRM 시스템 내부의 WSRF 표준을 적용한 웹 서비스 기반의 GNSI 인터페이스를 구현하기 위하여 Apache Muse Core를 사용하였고, 웹 기반의 HMI를 구현하기 위하여 JSP와 Apache Tomcat 5.5를 사용하였다. 본 네트워크 자원 할당 시험에서는 인터넷 익스플로러를 이용하여 HMI의 각 페이지를 시험한 결과를 기술하였다.

3.3 네트워크 자원의 생성 및 삭제

웹 기반의 HMI에서 네트워크 자원관리시스템을 통해 GMPLS 네트워크 경로 및 대역폭 관리를 위한 GNP, GNR, GRR 생성 및 자원 예약 시험 테스트 결과는 다음과 같다.

3.3.1 GNP 네트워크 경로 생성

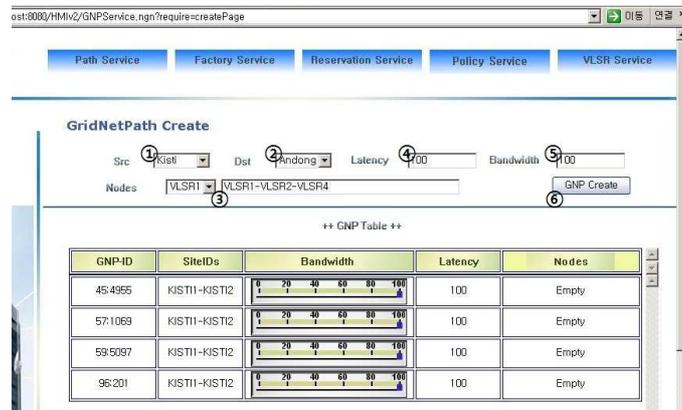


Figure 6. GNP Creation GUI

화면구성	<ol style="list-style-type: none"> ① GNP를 생성하기 위한 발신지 정보 선택 ② GNP를 생성하기 위한 목적지 정보 선택 ③ 발신지에서 목적지까지의 GNP 생성 시 경유할 중간 노드들을 입력 ④ GNP에서 허용되는 최대 지연의 정보를 입력 ⑤ GNP에서 허용되는 최대 대역폭의 정보를 입력 ⑥ GNP의 생성을 요청하는 버튼
입력값	<ol style="list-style-type: none"> ①②③④⑤⑥ ⑥ 클릭 시 GNP 요청 메시지 생성
이벤트	<ul style="list-style-type: none"> - ⑥ 클릭 시 GNP 생성을 위한 요청을 한다. - GNP 생성이 성공되면 그림 7과 같이 팝업 창으로 GNP 생성의 성공에 대한 메시지를 출력한다.

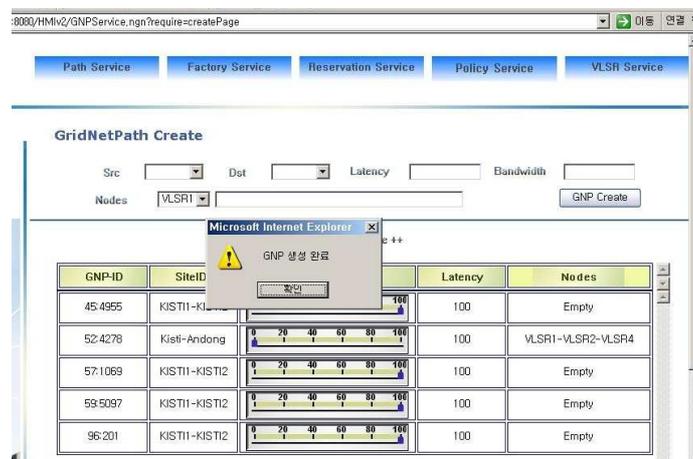


Figure 7. GNP Creation Result GUI

3.3.2 GNR 네트워크 자원 생성

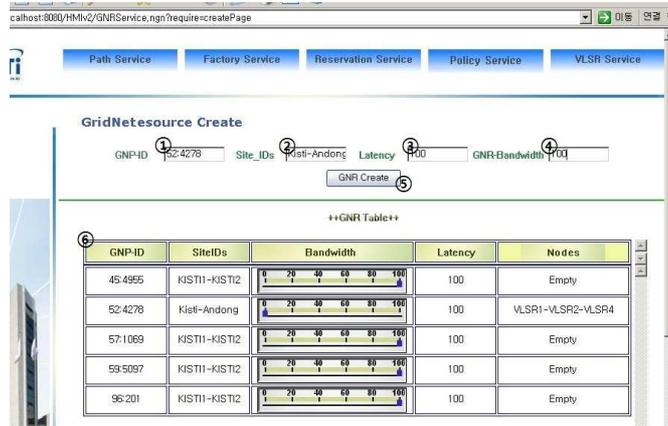


Figure 8. GNR Creation GUI

화면구성	<ul style="list-style-type: none"> ① 생성할 GNR의 GNP- ID를 입력 ② 생성할 GNR의 사이트 ID들을 입력 ③ 생성할 GNR에서 허용하는 지연을 입력 ④ 생성할 GNR에서 지원하는 최대 대역폭을 입력 ⑤ GNR의 생성을 요청하는 버튼 ⑥ GNP 테이블의 정보요소와 동일
입력값	①②③④
출력값	⑤ 클릭 시 GNR 요청 메시지 생성
이벤트	<ul style="list-style-type: none"> - 해당 GNP 선택 시 ①②③에 관련된 정보가 입력된다. - ⑤ 클릭 시 GNR의 생성에 대한 요청을 한다. - GNR 생성 결과를 그림 9와 같이 팝업 창으로 나타낸다.

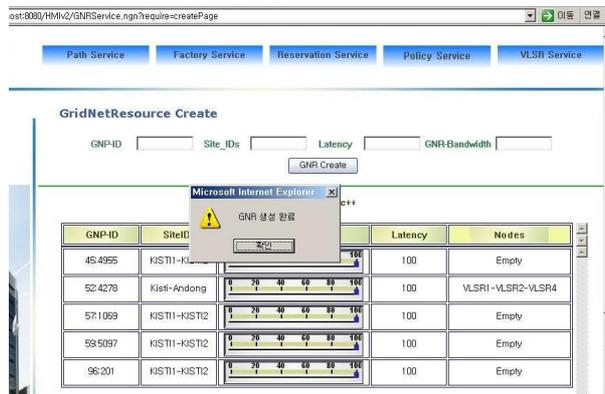


Figure 9. GNR Creation Result GUI

3.3.3. GRR 네트워크 자원 예약

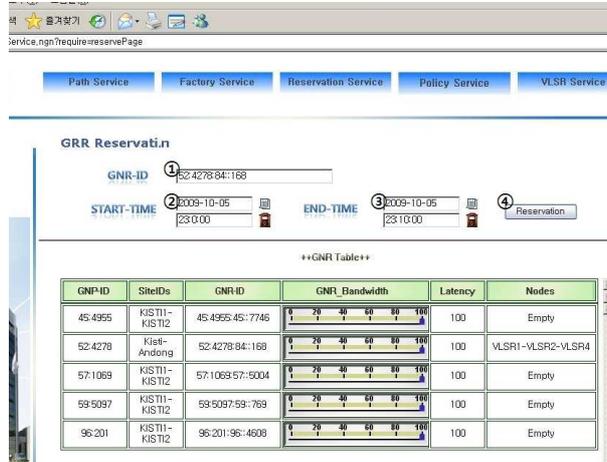


Figure 10. GRR Reservation GUI

화면구성	① 예약하고자 하는 자원의 식별자를 입력 ② 예약 기반의 자원예약 시작 날짜와 시간을 입력 ③ 예약 기반의 자원예약 종료 날짜와 시간을 입력 ④ 자원예약을 요청하는 위한 버튼
입력값	①②③④
출력값	①②③④
이벤트	- GNR 테이블에서 예약하려는 GNR을 클릭하면 ①에 해당 GNR- ID가 입력된다. - ④ 클릭 시 자원 예약을 요청한다. - 예약이 완료되면 그림 11과 같이 자원 예약 결과를 팝업 창으로 나타낸다.



Figure 11. GRR Reservation Result GUI

마지막 GNP, GNR, GRR을 통해 네트워크 자원의 예약이 완료되어지면 예약한 날짜와 시간에 예약한 대역폭 만큼의 네트워크 자원이 중단간 CSA상의 제어평면이 관리하는 데이터

평면 경로상에 보장되어 지며 ping 테스트는 예약한 시간동안 성공하게 된다.

4. 결론

최근 선진 연구망은 대용량의 트래픽을 요구하는 e-Science 응용과 같은 Community 들에게 supercomputers, computational clusters, storage array, 가시화 장비 등 네트워크 연계 자원들을 네트워크 자원 (대역폭, 경로)과 동시에 동적으로 제공해 줄 수 있는 융합 연구망으로의 변화를 꾀하고 있다. 그것을 실현해 주기 위해 네트워크 측면에서는 먼저 heterogeneous한 네트워크 환경에서 사용자가 요구시 동적으로 연결 지향형 네트워크를 제공해 줄 수 있어야 하며, 그와 더불어 사용자가 직접 네트워크 자원을 할당 제어해 줄 수 있는 네트워크 자원 할당 및 관리를 위한 서비스 기술이 필수적이라 말할 수 있다. Heterogeneous한 네트워크 환경에서 동적으로 연결 지향형 네트워크를 구성하기 위해 GMPLS 제어평면 기술은 가장 보편적인 기술이라 할 수 있으며 본 보고서에서는 DRAGON SW를 활용한 GMPLS 테스트베드망 상에서의 네트워크 자원의 관리 및 할당 시험을 기술하였다.

DRAGON은 GMPLS를 지원하지 않는 네트워크와 GMPLS 네트워크를 관통하여 하나의 LSP를 만들어 상이한 네트워크 기술 간에 통합된 제어 메커니즘을 제공한다. 이것은 안정적이고 보장된 대역폭을 보장받기를 원하는 e-Science 등 융합 응용들의 협업 연구에 좀 더 유연한 네트워크 환경을 제공한다. 국가 연구망인 KREONET은 GMPLS로의 진화를 앞두고 있으나, 현재는 GMPLS를 지원하는 네트워크 장비가 없다. 이러한 이유로 실제 GMPLS 네트워크망이 아닌 DRAGON SW를 활용하여 GMPLS 테스트베드 망을 구현하여 보고 사용자가 직접 네트워크 관리시스템을 통해 원하는 시간, 대역폭 및 경로를 예약하는 네트워크 자원의 동적 할당을 시험하게 된 것이다. 이를 위해 먼저 DRAGON의 구조를 분석하였고, 각 DRAGON component (CSA, VLSR, NARB)가 인트라 도메인 상에서의 GMPLS 제어평면을 지원하도록 각 시스템의 설정 파일을 수정하였다. DRAGON SW를 활용한 GMPLS 테스트베드망의 데이터평면은 Cisco 3750 이더넷 스위칭 장비를 활용하였고, 결국 제어평면의 VLSR이 데이터 평면의 이더넷 스위칭 장비의 VLAN을 제어 하였다. VLSR은 장비가 추가될 때 마다 모듈들의 수정이 불가피 하지만 DRAGON에서 모듈의 수정 없이도 지원이 되는 cisco3750 스위치를 사용하여 별도의 모듈 수정은 필요하지 않았다.

구현된 GMPLS 테스트베드망 상에서 e-Science와 같은 융합망 어플리케이션의 네트워크 자원 할당을 위해 book-ahead 모드 3계층 (GNP, GNR, GRR) 기반의 예약 할당 방식을 적용한 네트워크 관리시스템 (NRM)을 활용하였다. NRM은 외부 인터페이스로 WSRF 기반의 GNSI, 웹기반의 HMI 그리고 GMPLS 제어망 인터페이스인 GUNI를 갖는다. GNRM은 GNSI 및 HMI 메시지를 처리하는 GNMH 블록, 웹기반의 관리자 인터페이스 지원하는 HMIH 블록, 경로의 계산과 자원 예약 및 수락기능을 수행하는 PCRA 블록 그리고 GMPLS 제어망과의 인터페이스를 처리하는 GUMH 블록으로 구성된다. 구현된 NRM 시스템을 통해 GMPLS 테스트베드망과 연동하여 HMI 기반으로 네트워크 자원에 대한 관리 및 할당을 최종 시험하였다.

향후 과제로는 KREONET이 실제 GMPLS를 지원하는 망으로 전환 하였을 시를 대비하여 DRAGON에 종속적이 아닌 실제 GMPLS 제어평면과의 인터페이스 기능을 갖춘 그리고 인터도메인 환경에서도 적용 할 수 있는 NRM을 확장 개발하는 것이다. 또한, 실제 GMPLS

테스트베드망 상에서 NRM과의 연동을 통해 네트워크 자원의 관리 및 할당이 가능한지를 검증해야 할 것이다.

참 고 문 헌

- [1] "Virtual Label Switching Router Implementation Guide, version 2.1a" December, 2007.
- [2] "NARB and RCE Architecture", December 2007, <http://dragon.east.isi.edu>
- [3] "NARB Design and User Manual", December, 2007, <http://dragon.east.isi.edu>
- [4] "RCE Design and User Manual", December 2007, <http://dragon.east.isi.edu>
- [5] "GNU Zebra Routing Protocol Suite", <http://www.zebra.org>
- [6] Mark Meijerink, Rob Prickaerts, "Generalized MPLS, The DRAGON Project implementation at SARA", July 2006.
- [7] "DRAGON Workshop Lab Exercises", Designing and Engineering Dynamic Circuit Services:A Hands-On Workshop, 2007
- [8] IETF CCAMP, "<http://www.ietf.org/html.charters/ccamp-charter.html>" Dynamic Resource Allocation via GMPLS Optical Networks, "<http://dragon.east.isi.edu>"
- [9] Ibrahim W. Harbib, Q. Song, Z. Li, and Nageswara S. V. Rao, 'Deployment of the GMPLS Control Plane for Grid Applications in Experimental High-Performance Networks', IEEE Communications Magazine, pp. 65-73, March 2006.
- [10] M Hayashi, T Miyamoto, T Otani, H Tanaka, A Takefusa, H Nakada, T Kudoh, N Nagatsu, U Sameshima, S Oamoto, "Managing and controlling GMPLS network resources for Grid applications," OWQ,3, OFC 2006
- [11] A Takefusa, M Hayashi, n Nagatsu, H Nakada, T Kudoh, T Miyamoto, T Otani, H tanaka, M Suzuki, Y Sameshima, W Imajuku, M Jinno, Y takigawa, s Okamoto, Y Tanaka, S Sekiguchi, "G-lambda : Coordination of a grid scheduler and lambda path service over GMPLS," Future Generation Computing Systems, 2006
- [12] G. Markidis, A. Tzanakaki, N. Ciulli, G. Carrozzo, D. Simeonidou, R. Nejabati, G. Zervas, "EU Integrated Project PHOSPHORUS: Grid-GMPLS Control Plane for the Support of Grid Network Services," ICTON 2007
- [13] Xi Yang, Tom Lehman and Chris Tracy, "Policy-based Resource Management and Service Provisioning in GMPLS Networks," INFOCOM 2006.
- [14] Erik-Jan Bos, Gigi Karmous-Edwards, "GLIF Universal Service Interface GUSI A GLIF Project," GLIF 2008.
- [15] G-Lambda, "Inter-domain advance reservation of coordinated network and computing resources over the Pacific," GLIF 2007
- [16] A Takefusa, M Hayashi, n Nagatsu, H Nakada, T Kudoh, T Miyamoto, T Otani, H tanaka, M Suzuki, Y Sameshima, W Imajuku, M Jinno, Y takigawa, s Okamoto, Y Tanaka, S Sekiguchi, "G-lambda : Coordination of a grid scheduler and lambda path service over GMPLS," Future Generation Computing Systems, 2006
- [17] The G-lambda project, "Grid Network Service-Web Services Interface, version 3," OGF, 2008