

ISBN 978-89-6211-626-7

기술매뉴얼

융합망 통합자원관리시스템 연동을 위한
GMPLS 테스트베드망 개발

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

융합망 통합자원관리시스템 연동을 위한 GMPLS 테스트베드망 개발

임 헌 국, 문 정 훈, 공 정 욱

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

최종 수정일 2010년 11월 10일

Abstract

최근 선진 연구망은 의료, 바이오, 항공우주, e-Science 응용과 같은 첨단 응용 연구자들에게 supercomputers, clusters, storage, 가시화 장비 등과 같은 연구 자원들을 네트워크 자원 과 동시에 동적으로 제공해 줄 수 있는 융합 연구망으로의 변화를 꾀하고 있다. 이에 우리의 연구망도 그러한 변화에 발맞추어 2008년도부터 융합망 사업을 추진하고 있으며, 융합망 환경을 실현해 주기 위해 다음과 같은 요소 기술을 개발하고 있다. 먼저 네트워크 제공 측면에서는 heterogeneous한 네트워크 환경에서 사용자가 요구시 동적으로 연결 지향형 네트워크를 제공해 줄 수 있는 GMPLS 테스트베드 망과, 그와 더불어 사용자가 직접 네트워크 자원 및 연구 자원을 예약, 제어해 줄 수 있는 통합자원관리시스템 기술을 개발하고 있다. Heterogeneous한 네트워크 환경에서 동적으로 연결 지향형 네트워크를 구성하기 위해 GMPLS 제어평면 기술은 가장 보편적인 기술이라 할 수 있다.

본 보고서에서는 지역적으로 분산된 연구망 컴퓨팅 자원과 네트워크 자원의 통합 관리 및 예약 할당을 시험해 보기하기 위해, GMPLS 제어평면 기술을 제공하는 주니퍼사의 라우터를 활용하여 3계층 GMPLS 테스트베드 망을 개발하여 보고 구현된 GMPLS 테스트베드망 상에서 융합망 통합자원관리시스템을 통해 네트워크 및 컴퓨팅 자원의 예약 및 관리가 가능한지를 검증 하였다. 또한 최종적으로 네트워크자원관리시스템을 통해 전달되어야 하는 라우터 GMPLS config 사항을 기술하였고, GMPLS 경로를 통해 실제 트래픽이 흘러가는지를 기본적으로 테스트하였다.

Topics

1. 개요
2. 융합망 통합자원관리시스템
3. 통합자원관리시스템 연동을 위한 GMPLS 테스트베드 망
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 사용자가 직접 네트워크 자원 및 컴퓨팅, 스토리지 자원 등을 함께 할당 관리해 줄 수 있는 통합자원관리 시스템 기술이 제공되어야 하며, 또한 Heterogeneous한 네트워크 환경에서 동적으로 연결 지향형 네트워크를 구성하기 위해 GMPLS 제어평면이 하부 네트워크에서 필수적으로 제공되어야 한다. 이에 우리의 연구망도 2008년도부터 현재까지 융합망 사업을 통해 GMPLS 네트워크 구축 및 통합자원관리시스템을 개발하고 있다.

본 보고서에서는 GMPLS 테스트베드망 상에서의 융합망 자원의 관리 및 할당을 위한 통합자원관리시스템 연동을 위해, 주니퍼 네트워크 장비인 T640, T320, MX480, MX240 라우터 장비를 활용하여 3계층 GMPLS 테스트베드 망을 개발하여 보고 구현된 GMPLS 테스트베드망 상에서 융합망 통합자원관리 시스템을 통해 각 시스템 (GRS, NRM, CRM) 의 기본 기능을 검증 하였다. 사용되어진 각 T640, T320, MX480, MX240 라우터 장비는 주니퍼에서 제공되어진 JUNOS 9.1R1.8 version OS가 탑재되었다. 그리고 GMPLS config의 하나의 예로서 서울 본원에 위치한 T640 라우터에 연결되어있는 클러스터와 대전 본원의 통신실에 위치한 T320라우터에 연결되어 있는 클러스터, 그리고 T320라우터에 연결되어 있는 클러스터와 대전 본원 Antlab의 MX240에 연결되어 있는 클러스터를 각각 FEC 호스트 기반으로 config 하였다. 이를 통해 최종적으로 네트워크자원관리시스템을 통해 전달되어야 하는 라우터 GMPLS config 사항을 확인 할 수 있었고, GMPLS 경로를 통해 실제 트래픽이 흘러가는지를 기본적으로 테스트하였다.

2. 융합망 통합자원관리시스템

2.1 융합망 개발 배경 및 필요성

융합망의 개발 필요성은 광의의 의미와 협의의 의미로 나뉘어 생각해 볼 수 있다. 먼저 광의의 의미는 다음과 같다. 현재의 연구망은 첨단 R&D 협업연구를 위해 필요한 네트워크 자원을 수동적으로 지원하는 형태이므로 많은 예산이 소요될 뿐 아니라 응용 연구자 요구 기반으로 자원을 지원해 주기에는 한계가 있다. 따라서 미래형 연구망 환경인 융합망으로 전환하여 첨단 R&D 활동을 촉진하고 미래 수요를 조기에 창출하는 것을 필요로 한다. 미래형 연구망환경이란 신물질 개발 연구, 초대형 연구, 나노, 바이오 등 복잡하고 고난이도의 과학기술 응용 연구 활동이 가능하도록 응용 연구자의 요구에 알맞은 자원들을 융합하여 제공해 줄 수 있는 현재의 연구망과는 차별화된 새로운 유형의 네트워크 환경을 의미한다. 따라서 위와 같은 필요성에 부합한 사이버 연구망 환경 제공을 위해 연구 자원을 동적으로 예약 할당 할 수 있는 인프라 구축과 자원의 통합 제어 및 관리가 가능한 통합자원관리시스템을 개발 하여 융합서비스를 제공하여야 한다.

협의의 의미의 필요성은 다음과 같다. 공급자가 자원을 제공해주는 정적인 서비스에서 벗어나 이제는 사용자가 직접 웹 포털을 통해 네트워크 자원 (대역폭, 예약시간, 경로)과 컴퓨팅자원 (CPU 개수, 예약시간)을 동시에 실시간으로 예약 관리함을 통해 연구망에 분산되어 있는 컴퓨팅 자원을 효율적으로 활용하며 응용 연구개발의 비용 절감을 이끌어 낼 필요가 있다.

2.2 융합망 통합자원관리시스템 구성요소

융합망 통합자원관리시스템은 아래 그림처럼 GMPLS 네트워크 기반위에서 동작한다. 시스템 구성 요소들은 사용자가 현재의 자원 현황을 직접 모니터링 하고 필요한 자원을 직접 예약 관리 하는데 필요한 사용자 웹 포털과 사용자로부터 전달된 두 가지 자원에 대한 입력정보 (조회, 예약, 해제/취소)를 컴퓨팅자원관리시스템과 네트워크자원관리시스템에게 전달하는 GRS 시스템, 그리고 각 자원에 대해 실제 조회, 예약 및 해제/취소, 할당 역할을 수행하는 컴퓨팅 자원관리시스템과 네트워크 자원관리시스템이 존재한다.

현재까지 개발된 상황을 살펴보면 다음과 같다. 먼저 GRS 작업(컴퓨팅/네트워크 자원을 동시에 활용) 제출을 위해 웹기반 사용자 인터페이스를 개발하였으며, GRS 작업 모니터링을 위한 웹기반 사용자 인터페이스를 개발하였다. 또한 국제 표준 규격 GNSI 기반의 웹서비스를 지원하는 NRM을 개발하였으며, 타임 테이블 기반의 컴퓨팅 자원 관리 시스템(CRM)을 개발 완료하였다. 현재까지 개발된 시스템들은 관리자모드의 시범서비스용이며 2011년 2012년 개발 안전화 단계를 거쳐 실제 사용자에게 서비스가 가능한 시스템으로 전환을 할 예정이다. 개발되어진 NRM(네트워크 자원 관리 시스템), CRM(컴퓨팅 자원 관리 시스템), GRS(글로벌 스케줄러)와 서울-대전 사이에 구현된 GMPLS 테스트베드 망 및 클러스터 자원과의 연동을 통해 각 시스템의 기본 기능을 테스트하였고 가장 기본적인 자원 예약 및 할당 시나리오에 따라 자원 할당 시연을 수행하였다.

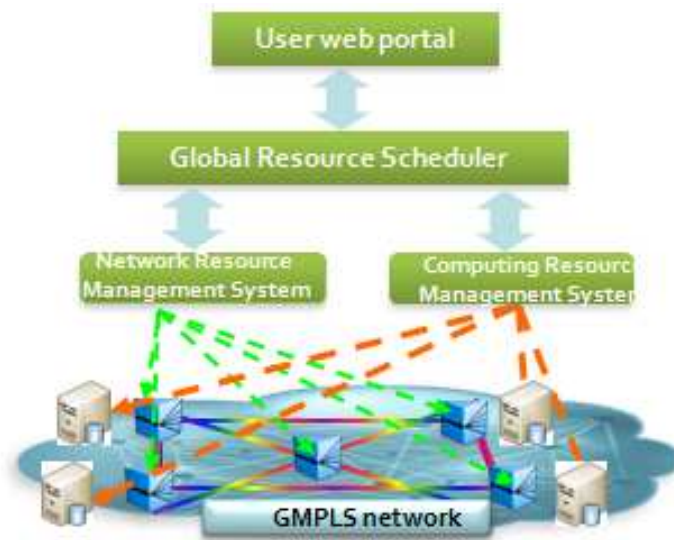


그림 1. 융합망 통합자원관리시스템 구성요소

3. 통합자원관리시스템 연동을 위한 GMPLS 테스트베드망

3.1 융합망 통합자원관리시스템 연동 GMPLS 테스트베드망 시험 시연 시나리오

융합망의 각 시스템 컴포넌트 (CRM, NRM, GRS)와 서울-대전 간 구축 개발 되어진 GMPLS 테스트베드망을 연동하여 자원할당 시험을 테스트하였다. 컴퓨팅 자원과 네트워크 자원의 동시 예약이 GRS 포털 시스템을 통해 CRM, NRM 을 통해 이루어졌으며 컴퓨팅 자원의 예약은 사용자가 원하고자하는 클러스터의 CPU 개수, 예약시간을 네트워크 자원에 대해서는 원하고자하는 경로, 대역폭, 예약시간이 예약되어진다. 네트워크 자원의 예약시간의 start time에 예약된 경로의 GMPLS LSP와 대역폭의 set-up이 예약시간의 end time에 예약된 경로의 GMPLS LSP와 대역폭이 release 되어 지며. 컴퓨팅 자원에 대해서는 예약 시간의 start time에 작업의 시작이 예약시간의 end time에 작업의 종료이 이루어진다. 만약 예약된 시간의 end time보다 작업이 먼저 끝나게 되면 선택된 CPU에 대해 release가 이루어진다. 각 클러스터 노드는 서울 키스티 본원의 T640 라우터에, 대전 키스티 본원의 통신실 T320 라우터에, 대전 키스티 본원 ANTlab의 MX240 라우터에 직접 연결되어 있으며, GMPLS LSP는 3계층으로 연동하였다. 아래 그림은 GMPLS 테스트베드망 지역 연동 구성도를 보여준다.

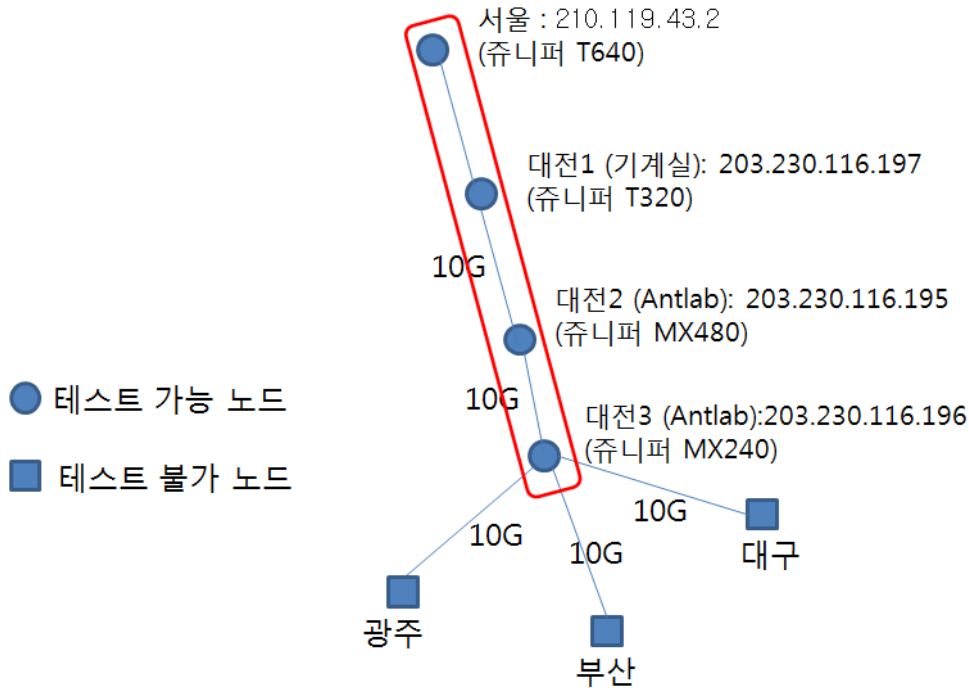


그림 2 GMPLS 테스트베드망 지역 연동 구성도

융합망 각 구성 컴포넌트들과 연동된 GMPLS 테스트베드망의 시험 검증을 위해 간단한 데모 시연을 하였으며, 그 시연의 시나리오는 다음과 같다. 서울 T640 라우터에 연결된 클러스터에 저장된 데이터를 대전의 MX240 라우터에 연결된 클러스터로 전송한 후 그곳 클러스터를 활용 컴퓨팅 작업을 완료하여 다시 서울의 T640 라우터로 결과 데이터를 전송하는 시나리오가 채택되었다. 아래 그림이 설명되어진 시연 시나리오이다.

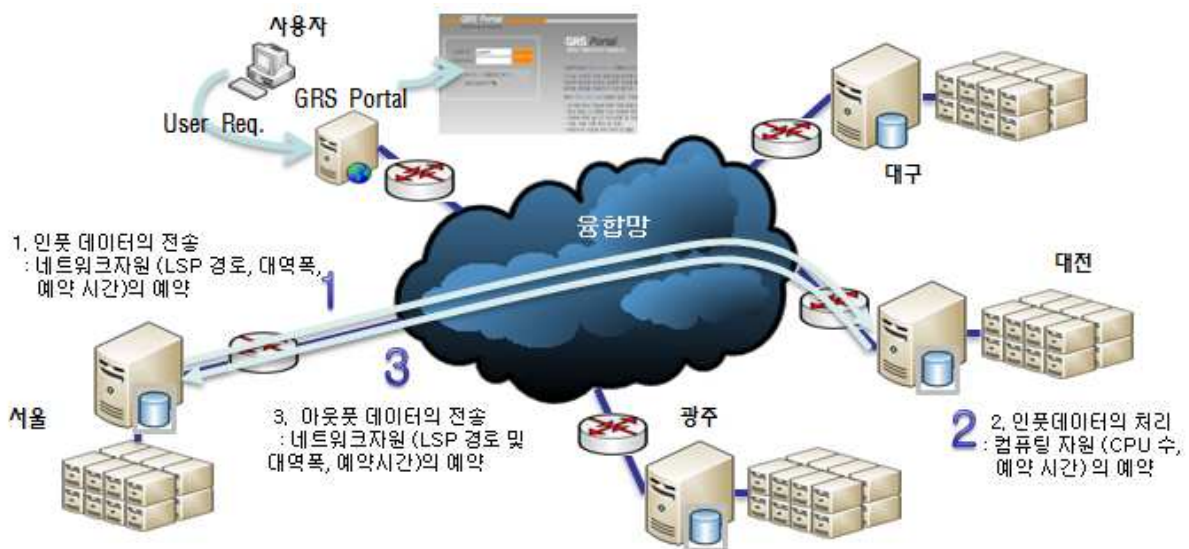


그림 3 통합자원관리시스템을 활용한 자원 예약 및 할당 시연 시나리오

3.2 라우터 configuration 사항 및 테스트 결과

각 라우터의 GMPLS 테스트베드망 세부 연동 구성도는 아래 그림과 같다. 클러스터는 서울 키스티 통신실의 T640 라우터에, 그리고 대전 키스티 통신실의 T320 라우터에 그리고 대전 키스티 ANTlab의 MX240 라우터에 각각 연동되어 있으며, 컴퓨팅 자원을 위한 예약으로 사용되어 진다. 각 라우터에 연동을 위한 configuration을 위해 외부접속 공인 IP를 ge 인터페이스에 하나씩 할당하였으며 각 라우터의 GMPLS 데이터평면 연동을 위해서는 사설 IP인 192.168.X.X 대역을 할당하였다. GRS는 아래 GMPLS 테스트베드망 위에 외부접속 공인 IP를 통해 각각 CRM (컴퓨팅 자원관리시스템), NRM (네트워크 자원관리시스템)의 외부접속 공인 IP와 서로 통신하며 통합자원관리 역할을 수행한다.

그리고 GMPLS config의 하나의 예로서 T640 라우터에 연결되어있는 클러스터와 T320 라우터에 연결되어 있는 클러스터, 그리고 T320라우터에 연결되어 있는 클러스터와 MX240에 연결되어 있는 클러스터를 FEC 호스트기반으로 config 하였다.

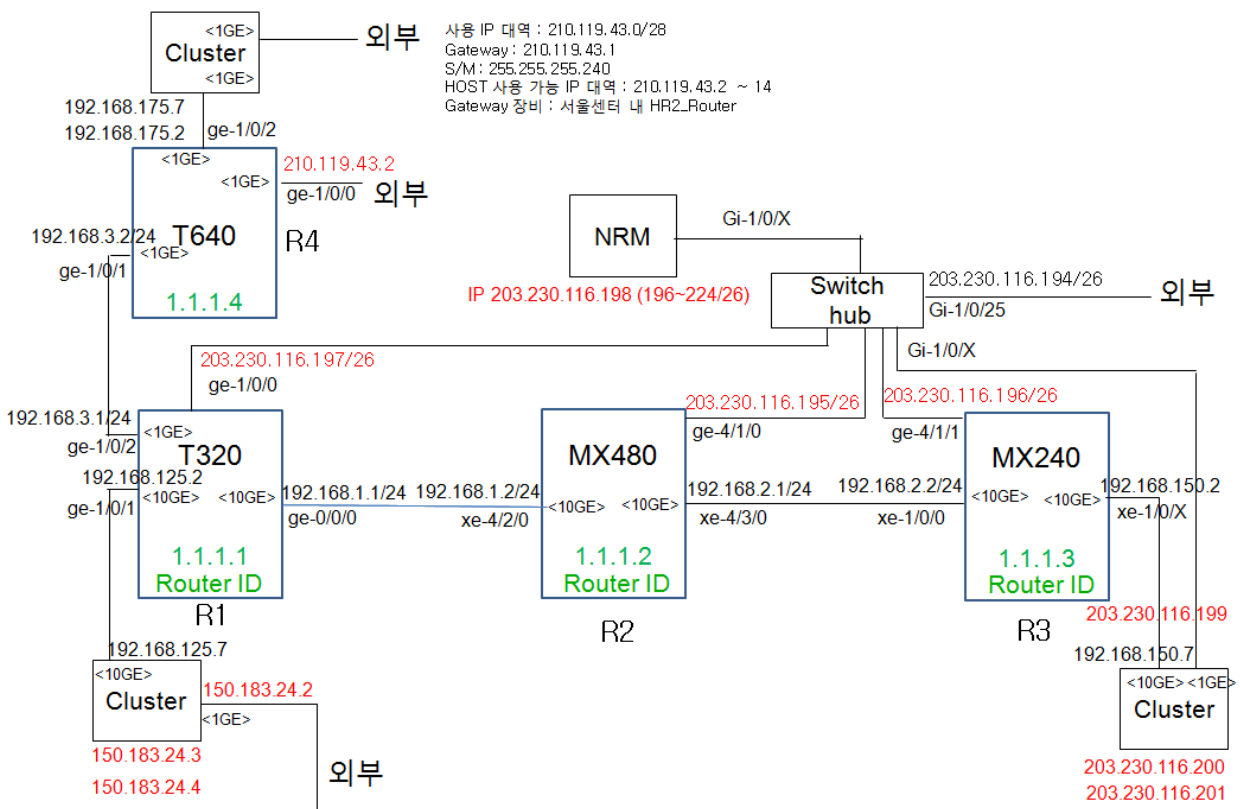


그림 4 GMPLS 테스트베드망 세부 연동 구성도

3.2.1 T640 라우터 configuration 사항 및 테스트 결과

먼저 서울 분원에 위치한 주니퍼 T640 라우터의 ospf 라우팅 테이블이 제대로 올라가 있는지 확인하였다. 아래 명령어를 활용하여 T640 라우터의 ospf 라우팅 테이블을 확인하도록 한다. GMPLS의 동작전에 ospf 라우팅 프로토콜을 이용하여 라우터와 라우터 그리고 클러스터간에 LSA (link state advertisement) 메시지를 이용하여 전체 네트워크의 link state를 확인하기 때문이다.

```
kisti@HR_T640> show route protocol ospf
inet.0: 19 destinations, 20 routes (19 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both
```

```
1.1.1.1/32      [OSPF/10] 2w2d 02:08:17, metric 1
                > to 192.168.3.1 via ge-1/0/1.0
1.1.1.2/32      *[OSPF/10] 1w2d 14:51:59, metric 2
                > to 192.168.3.1 via ge-1/0/1.0
1.1.1.3/32      *[OSPF/10] 1w2d 14:51:36, metric 3
                > to 192.168.3.1 via ge-1/0/1.0
192.168.1.0/24  *[OSPF/10] 1w2d 14:51:59, metric 2
                > to 192.168.3.1 via ge-1/0/1.0
192.168.2.0/24  *[OSPF/10] 1w2d 14:51:36, metric 3
                > to 192.168.3.1 via ge-1/0/1.0
192.168.125.0/24 *[OSPF/10] 2w2d 02:08:17, metric 2
                > to 192.168.3.1 via ge-1/0/1.0
192.168.150.0/24 *[OSPF/10] 5d 01:11:49, metric 4
                > to 192.168.3.1 via ge-1/0/1.0
203.230.116.192/26 *[OSPF/10] 1w2d 14:53:30, metric 2
                > to 192.168.3.1 via ge-1/0/1.0
224.0.0.5/32    *[OSPF/10] 33w5d 14:39:46, metric 1
                MultiRecv
```

```
inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)
__juniper_private1__inet.0: 2 destinations, 3 routes (2 active, 0 holddown, 0 hidden)
__juniper_private2__inet.0: 1 destinations, 1 routes (0 active, 0 holddown, 1 hidden)
mpls.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
__juniper_private1__inet6.0: 5 destinations, 7 routes (5 active, 0 holddown, 0 hidden)
```


T640 라우터에서 확인한 ospf 라우팅 테이블의 내용은 다음과 같았다. T640 라우터 자신의 인터페이스에 연동된 라우터 및 클러스터를 제외한 모든 라우터 및 클러스터의 ip address (192.168.1.0, 192.168.2.0, 192.168.125.0, 192.168.150.0) 와 연동되어진 모든 라우터의 루프백 어드레스 (1.1.1.1, 1.1.1.2, 1.1.1.3)를 확인할 수 있었다. 이를 이용하여 각 라우터는 네트워크의 토폴로지 및 상태 정보를 미리 파악할 수 있다.

다음으로 GMPLS configuration을 위해 필요한 사항들을 정리해 본다. 먼저 아래 명령을 이용하여 인터페이스 사항을 점검한다.

```
kisti@HR_T640> show configuration interfaces
```

```
interfaces {
  ge-0/0/0 {
    mtu 9192;
    unit 0 {
      family inet;
    }
  }
  ge-0/1/0 {
    unit 0 {
      family inet;
    }
  }
  ge-1/0/0 {
    unit 0 {
      family inet {
        address 210.119.43.2/28;
      }
    }
  }
  ge-1/0/1 {
    mtu 2000;
    unit 0 {
      family inet {
        address 192.168.3.2/24;
      }
      family mpls;
    }
  }
  ge-1/0/2 {
    unit 0 {
      family inet {
```

```

        address 192.168.175.2/24;
    }
    family mpls;
}
fxp0 {
    unit 0 {
        family inet {
            address 10.10.10.4/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 1.1.1.4/32;
        }
    }
}
}

```

먼저 T640 라우터 ge-1/0/0 인터페이스에 외부접속 및 관리를 위한 ip address로 210.119.43.2 번을 할당하였고 ge-1/0/1 인터페이스는 T320 라우터와의 연동을 위한 사설 ip address로 192.168.3.2 번을 할당하였다. 키스티 서울 본원의 클러스터 연동을 위해 ge-1/0/2 인터페이스에 192.168.175.2 번을 할당하였고 라우터의 루프백 어드레스 (router id)로 1.1.1.4 번을 할당하였다.

GMPLS 동작과 관련된 routing options 을 위해 특별히 config 해주어야 할 사항은 따로 존재하지 않으며 아래 명령을 이용하여 routing options을 점검해 보기로 한다. 인터넷 접속을 위한 next-hop을 게이트웨이 address인 210.119.43.1을 할당하였다.

```

kisti@HR_T640> show configuration routing-options
routing-options {
    static {
        route 0.0.0.0/0 next-hop 210.119.43.1;
    }
    router-id 1.1.1.4;
}

```

다음으로 GMPLS 동작과 관련된 protocols의 config 사항을 점검해 보기로 한다. 쥬니퍼에서 제공한 T series, MX series 라우터는 기본적으로 GMPLS 프로토콜을 제공하지만 3

계층에서는 아직 대역폭을 보장하기 위한 기능을 제공하지 않는다. 따라서 MPLS 프로토콜을 사용하여 양방향으로 LSP와 대역폭을 보장해주는 방식으로 GMPLS 프로토콜을 구현 하였다. 하지만 MPLS 프로토콜에서는 기본적으로 in-band 시그널링을 사용하기 때문에 RSVP 프로토콜 구현을 위해서 인터페이스는 데이터 플레인과 동일한 인터페이스를 사용해야 한다. 따라서 RSVP config에서 시그널링을 위한 인터페이스로 ge-1/0/1 를 사용하였다. 아래 명령을 사용하여 GMPLS protocol config와 관련된 사항을 점검해 보기로 한다.

```
kisti@HR_T640> show configuration protocols
protocols {
  rsvp {
    interface ge-1/0/1.0;
    interface lo0.0;
  }
  mpls {
    traffic-engineering bgp-igp-both-ribs;
    label-switched-path 1:2884:1::9750:3::3343 {
      to 1.1.1.1;
      install 192.168.125.7/32;
      lsp-attributes {
        switching-type psc-1;
        encoding-type packet;
      }
      bandwidth 5m;
      no-cspf;
      primary 1:2884:1::9750:3::3343;
    }
    path 1:2884:1::9750:3::3343 {
      1.1.1.1 loose;
    }
    interface all;
    interface fxp0.0 {
      disable;
    }
  }
  ospf {
    area 0.0.0.0 {
      interface ge-1/0/1.0;
      interface lo0.0 {
        passive;
      }
      interface ge-1/0/2.0 {
```

```

        passive;
    }
}
}
}

```

대전 통신실에 위치한 T320 라우터와의 GMPLS config를 위해 먼저 T640 라우터에서는 단방향 MPLS config를 필요로 한다. T320 라우터에 연결된 클러스터까지 FEC 호스트 기반의 설정을 위해 `install 192.168.125.7` 명령을 필요로 하며, 대역폭 설정을 위해 `bandwidth 5m` 라는 명령어를 활용하여 5M 대역폭을 보장했으며, 그리고 경로를 사용자가 직접 지정하고자 할 경우 `no-cspf` 라는 명령어를 사용하여 config 하는 것을 필요로 한다. label-switched-path 경로는 `1:2884:1::9750:3::3343` 이라는 NRM을 통해 전달받은 GRR ID를 사용하였다.

다음으로 GMPLS config를 통해 실제 lsp가 생성되었는지를 확인하기로 한다. 아래 명령을 사용하면 Ingress 라우터로서, 그리고 Egress 라우터로서, 그리고 Trnsnit 라우터로서 생성된 lsp 세션의 정보를 보여준다. 첫 번째 결과가 앞서 config를 통해 생성된 Ingress lsp 세션이며, 두 번째 Egress lsp 세션의 결과는 상대방 라우터 즉 T320 라우터에서 T640 라우터 쪽으로의 lsp 설정을 위해 T320 라우터에서 MPLS config한 것에 대한 결과이다. 마지막으로 transit 라우터로서 lsp 세션은 없음을 확인할 수 있다.

```
kisti@HR_T640> show mpls lsp
```

```
Ingress LSP: 1 sessions
```

To	From	State	Rt	ActivePath	P	LSPname
1.1.1.1	1.1.1.4	Up	2	1:2884:1::9750:3::3343	*	1:2884:1::9750:3::3343

```
Total 1 displayed, Up 1, Down 0
```

```
Egress LSP: 1 sessions
```

To	From	State	Rt	Style	Labelin	Labelout	LSPname
1.1.1.4	1.1.1.1	Up	0	1 FF	3	-	1:2884:1::9750:3::3343

```
Total 1 displayed, Up 1, Down 0
```

```
Transit LSP: 0 sessions
```

```
Total 0 displayed, Up 0, Down 0
```

마지막으로 ip 라우팅이 아닌 실제 설정된 lsp 경로를 통해 패킷이 label swapping을 통해 전송되는지를 확인하였다. `ping 192.168.125.7 rapid count 1000` 명령어를 사용하여 T320 라우터에 연결된 클러스터 까지 1000개의 패킷을 전송해 보았다.

```

kisti@HR_T640> ping 192.168.125.7 rapid count 1000
PING 192.168.125.7 (192.168.125.7): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
--- 192.168.125.7 ping statistics ---
1000 packets transmitted, 1000 packets received, 0% packet loss
round-trip min/avg/max/stddev = 2.902/3.243/74.438/2.866 ms

```

lsp를 통해 실제 패킷이 전송되는 것을 확인해 보기 위해서는 show mpls lsp statistics 명령을 이용하면 된다. 아래 결과처럼 1000개의 패킷이 lsp 경로를 통해 T320라우터에 연결된 클러스터까지 전송됨을 확인하였다.

```

kisti@HR_T640> show mpls lsp statistics
Ingress LSP: 1 sessions
To          From      State   Packets      Bytes LSPname
1.1.1.1     1.1.1.4   Up      1000         84000 1:2884:1::9750:3::3343
Total 1 displayed, Up 1, Down 0

Egress LSP: 1 sessions
To          From      State   Packets      Bytes LSPname
1.1.1.4     1.1.1.1   Up      NA           NA 1:2884:1::9750:3::3343
Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

```

3.2.2 T320 라우터 configuration 사항 및 테스트 결과

먼저 T320 라우터의 인터페이스 사항을 다음과 같이 설정하였다. 먼저 T320 라우터의 ge-1/0/0 인터페이스에 외부접속 및 관리를 위한 ip address로 203.230.116.197 번을 할당하였고 ge-1/0/2 인터페이스는 T640 라우터와의 연동을 위한 사설 ip address로 192.168.3.1 번을 할당하였다. 키스티 대전 ANTIlab의 클러스터 연동을 위해 ge-1/0/2 인터페이스에 192.168.125.2 번을 할당하였고 라우터의 루프백 어드레스 (router id)로 1.1.1.1 번을 할당하였다. 마지막으로 MX480 라우터와의 연동을 위한 인터페이스로 ge-0/0/0 에 192.168.1.1 번 사설 ip address를 할당하였다.

```
kisti@HR_T640> show configuration interfaces
```

```
interfaces {
  ge-0/0/0 {
    mtu 2000;
    unit 0 {
      family inet {
        address 192.168.1.1/24;
      }
      family mpls;
    }
  }
  ge-0/1/0 {
    unit 0 {
      family inet {
        address 192.168.125.2/24;
      }
      family mpls;
    }
  }
  ge-1/0/0 {
    unit 0 {
      family inet {
        address 203.230.116.197/26;
      }
    }
  }
  ge-1/0/2 {
    mtu 2000;
    unit 0 {
      family inet {
        address 192.168.3.1/24;
      }
    }
  }
}
```

```

    }
    family mpls;
  }
}
fxp0 {
  unit 0 {
    family inet {
      address 10.10.10.1/24;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 1.1.1.1/32;
    }
  }
}
}
}

```

GMPLS 동작과 관련된 routing options 을 위해 특별히 config 해주어야 할 사항은 따로 존재하지 않으며 아래 명령을 이용하여 routing options을 점검해 보기로 한다. 인터넷 접속을 위한 next-hop을 게이트웨이 address인 203.230.116.193을 할당하였다.

```

routing-options {
  static {
    route 0.0.0.0/0 next-hop 203.230.116.193;
  }
  router-id 1.1.1.1;
}

```

다음으로 GMPLS 동작과 관련된 protocols의 config 사항을 점검해 보기로 한다. 유니퍼에서 제공한 T series, MX series 라우터는 기본적으로 GMPLS 프로토콜을 제공하지만 3계층에서는 아직 대역폭을 보장하기 위한 기능을 제공하지 않는다. 따라서 MPLS 프로토콜을 사용하여 양방향으로 LSP와 대역폭을 보장해주는 방식으로 GMPLS 프로토콜을 구현 하였다. 하지만 MPLS 프로토콜에서는 기본적으로 in-band 시그널링을 사용하기 때문에 rsvp 프로토콜 구현을 위해서 인터페이스는 데이터 플레인과 동일한 인터페이스를 사용해야 한다. 따라서 rsvp config에서 시그널링을 위한 인터페이스로 ge-1/0/2 과 ge-0/0/0 를 사용 하였다. 아래 명령을 사용하여 GMPLS protocol config와 관련된 사항을 점검해 보기로 한다.

대전 통신실에 위치한 T640 라우터와의 GMPLS config를 위해 먼저 T320 라우터에서는 단방향 MPLS config 설정을 필요로 한다. T640 라우터에 연결된 클러스터까지 FEC 호스트 기반의 설정을 위해 install 192.168.175.7 명령을 필요로 하며, 대역폭 설정을 위해 bandwidth 5m 라는 명령어를 활용하여 5M 대역폭을 보장했으며, label-switched-path 경로는 1:2884:1::9750:3::3343 이라는 NRM을 통해 전달받은 GRR ID를 사용하였다.

MX240 라우터까지의 GMPLS 양방향 lsp 설정을 위해 T320 라우터에서는 역시 단방향 MPLS config 설정을 필요로 한다. mpls { } 안에 두 번째 label-switched-path 1:5979:1::9653:3::1658 config 설정 사항이 MX240 라우터까지의 MPLS config 설정 사항이며, MX240 라우터에 연결된 클러스터까지로 FEC 호스트 기반의 lsp 설정을 위해 install 192.168.150.7/32 명령을 역시 사용하였다.

```
protocols {
  rsvp {
    interface ge-0/0/0.0;
    interface lo0.0;
    interface ge-1/0/2.0;
  }
  mpls {
    traffic-engineering bgp-igp-both-ribs;
    label-switched-path 1:2884:1::9750:3::3343 {
      to 1.1.1.4;
      install 192.168.175.7/32;
      lsp-attributes {
        switching-type psc-1;
        encoding-type packet;
      }
      bandwidth 5m;
      no-cspf;
      primary 1:2884:1::9750:3::3343;
    }
    label-switched-path 1:5979:1::9653:3::1658 {
      to 1.1.1.3;
      install 192.168.150.7/32;
      lsp-attributes {
        switching-type psc-1;
        encoding-type packet;
      }
      bandwidth 5m;
      no-cspf;
      primary 1:5979:1::9653:3::1658;
    }
  }
}
```



```

    path 1:2884:1::9750:3::3343 {
        1.1.1.4 loose;
    }
    path 1:5979:1::9653:3::1658 {
        1.1.1.3 loose;
    }
    interface ge-0/0/0.0;
    interface lo0.0;
    interface ge-1/0/2.0;
}
ospf {
    area 0.0.0.0 {
        interface lo0.0 {
            passive;
        }
        interface ge-0/0/0.0;
        interface ge-1/0/0.0;
        interface ge-1/0/1.0;
        interface ge-0/1/0.0;
        interface ge-1/0/2.0;
    }
}
}

```

마지막으로 다른 연동 라우터와의 ospf 프로토콜 동작을 위해 T320 라우터의 ge-0/0/0, ge-1/0/1, ge-1/0/2 인터페이스를 ospf {} 안에 추가해 주어야 한다.

다음으로 GMPLS config를 통해 실제 lsp가 생성되었는지를 확인하기로 한다. 아래 명령을 사용하면 Ingress 라우터로서, 그리고 Egress 라우터로서, 그리고 Trnsnit 라우터로서 생성된 lsp 세션의 정보를 보여준다. 첫 번째 결과가 앞서 config를 통해 생성된 Ingress lsp 세션 2개이며, 두 번째 Egress lsp 세션의 결과는 상대방 라우터 즉 T320 라우터에서 T640 라우터 쪽으로의 lsp 설정을 위해 T320 라우터에서 MPLS config한 것에 대한 lsp 세션 1개와 뒷부분에서 설명하게 될 MX240 라우터에서 T320 라우터 쪽으로의 lsp 설정을 위해 MX240 라우터에서 MPLS config한 것에 대한 lsp 세션 1개를 합쳐 모두 2개의 session을 보여준다. 마지막으로 transit 라우터로서 lsp 세션은 없음을 확인할 수 있다.

```
kisti@DJ_T320_RE0> show mpls lsp
```

```
Ingress LSP: 2 sessions
```

To	From	State	Rt	P	ActivePath	LSPname
1.1.1.3	1.1.1.1	Up	3	*	1:5979:1::9653:3::1658	1:5979:1::9653:3::1658

```
1.1.1.4      1.1.1.1      Up    1 *    1:2884:1::9750:3::3343 1:2884:1::
9750:3::3343
```

Total 2 displayed, Up 2, Down 0

Egress LSP: 2 sessions

To	From	State	Rt	Style	Labelin	Labelout	LSPname
1.1.1.1	1.1.1.4	Up	0	1 FF	3	-	1:2884:1::9750:3::3343
1.1.1.1	1.1.1.3	Up	0	1 FF	3	-	1:5979:1::9653:3::1658

Total 2 displayed, Up 2, Down 0

Transit LSP: 0 sessions

Total 0 displayed, Up 0, Down 0

마지막으로 ip 라우팅이 아닌 실제 설정된 lsp 경로를 통해 패킷이 label swapping을 통해 전송되는지를 확인하였다. ping 192.168.150.7 rapid count 1000 명령어를 사용하여 MX240 라우터에 연결된 클러스터까지 1000개의 패킷을 전송해 보았다.

```
kisti@DJ_T320_RE0> ping 192.168.150.7 rapid count 1000
```

```
PING 192.168.150.7 (192.168.150.7): 56 data bytes
```

```
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
```

```
--- 192.168.150.7 ping statistics ---
```

```
1000 packets transmitted, 1000 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.278/0.360/25.016/1.032 ms
```

lsp를 통해 실제 패킷이 전송되는 것을 확인해 보기 위해서는 show mpls lsp statistics 명령을 이용하면 된다. 아래 결과처럼 1000개의 패킷이 lsp 경로를 통해 MX240 라우터에 연결된 클러스터까지 전송됨을 확인하였다.

```
kisti@DJ_T320_RE0> show mpls lsp statistics
```

```
Ingress LSP: 2 sessions
```

To	From	State	Packets	Bytes	LSPname
1.1.1.3	1.1.1.1	Up	1000	84040	1:5979:1::9653:3::1658
1.1.1.4	1.1.1.1	Up	0	0	1:2884:1::9750:3::3343

```
Total 2 displayed, Up 2, Down 0
```

```
Egress LSP: 2 sessions
```

To	From	State	Packets	Bytes	LSPname
1.1.1.1	1.1.1.4	Up	NA	NA	1:2884:1::9750:3::3343
1.1.1.1	1.1.1.3	Up	NA	NA	1:5979:1::9653:3::1658

```
Total 2 displayed, Up 2, Down 0
```

```
Transit LSP: 0 sessions
```

```
Total 0 displayed, Up 0, Down 0
```

3.3.3 MX480 라우터 configuration 사항 및 테스트

먼저 JUNIPER MX480 라우터의 ospf 라우팅 테이블이 제대로 올라가 있는지 확인하였다. 아래 명령어를 활용하여 T640 라우터의 ospf 라우팅 테이블을 확인하도록 한다. GMPLS의 동작전에 ospf 라우팅 프로토콜을 이용하여 라우터와 라우터 그리고 클러스터간에 LSA (link state advertisement) 메시지를 이용하여 전체 네트워크의 link state를 확인하기 때문이다.

```
kisti@MX480> show route protocol ospf
```

```
inet.0: 18 destinations, 19 routes (18 active, 0 holddown, 0 hidden)
```

```
+ = Active Route, - = Last Active, * = Both
```

```
1.1.1.1/32      *[OSPF/10] 1w2d 15:17:35, metric 1
                > to 192.168.1.1 via xe-4/2/0.0
1.1.1.3/32      *[OSPF/10] 1w2d 15:17:08, metric 1
                > to 192.168.2.2 via xe-4/3/0.0
1.1.1.4/32      *[OSPF/10] 1w2d 15:17:35, metric 2
                > to 192.168.1.1 via xe-4/2/0.0
192.168.3.0/24  *[OSPF/10] 1w2d 15:17:35, metric 2
                > to 192.168.1.1 via xe-4/2/0.0
192.168.125.0/24 *[OSPF/10] 1w2d 15:17:35, metric 2
                > to 192.168.1.1 via xe-4/2/0.0
```

```

192.168.150.0/24  *[OSPF/10] 5d 01:36:38, metric 2
                  > to 192.168.2.2 via xe-4/3/0.0
192.168.175.0/24  *[OSPF/10] 1w0d 00:07:54, metric 3
                  > to 192.168.1.1 via xe-4/2/0.0
203.230.116.192/26 [OSPF/10] 1w2d 15:17:35, metric 2
                  > to 192.168.1.1 via xe-4/2/0.0
224.0.0.5/32      *[OSPF/10] 1w2d 15:19:12, metric 1
                  MultiRecv

```

```

__juniper_private1__inet.0: 10 destinations, 12 routes (6 active, 0 holddown, 4
hidden)

```

```

__juniper_private2__inet.0: 1 destinations, 1 routes (0 active, 0 holddown, 1 h
idden)

```

```

mpls.0: 7 destinations, 7 routes (7 active, 0 holddown, 0 hidden)

```

```

__juniper_private1__inet6.0: 5 destinations, 7 routes (5 active, 0 holddown, 0
hidden)

```

MX480 라우터에서 확인한 ospf 라우팅 테이블의 내용은 다음과 같았다. MX480 라우터 자신의 인터페이스에 연동된 라우터 및 클러스터를 제외한 모든 라우터 및 클러스터의 ip address (192.168.3.0, 192.168.175.0, 192.168.125.0, 192.168.150.0) 와 연동되어진 모든 라우터의 루프백 어드레스 (1.1.1.1, 1.1.1.3, 1.1.1.4)를 확인할 수 있었다. 이를 이용하여 각 라우터는 네트워크의 토폴로지 및 상태 정보를 미리 파악할 수 있다.

다음으로 MX480 라우터의 인터페이스 사항을 다음과 같이 설정하였다. 먼저 MX480 라우터의 ge-4/1/0 인터페이스에 외부접속 및 관리를 위한 ip address로 203.230.116.195 번을 할당하였고 xe-4/2/0 인터페이스는 T320 라우터와의 연동을 위한 사설 ip address로 192.168.1.2 번을 할당하였다. 또한 xe-4/3/0 인터페이스는 MX240 라우터와의 연동을 위한 사설 ip address로 192.168.2.1 번을 할당하였다. 마지막으로 MX480 라우터의 루프백 어드레스 (router id)로 1.1.1.2 번을 할당하였다. 아래 명령어를 사용하여 인터페이스에 관련된 설정 사항을 확인하였다.

```

kisti@MX480> show configuration interfaces

```

```

interfaces {
  ge-4/0/1 {
    unit 0 {
      family inet;
    }
  }
}

```

```
ge-4/1/0 {
  unit 0 {
    family inet {
      address 203.230.116.195/26;
    }
  }
}
xe-4/2/0 {
  mtu 2000;
  unit 0 {
    family inet {
      address 192.168.1.2/24;
    }
    family mpls;
  }
}
xe-4/2/1 {
  unit 0 {
    family inet;
  }
}
xe-4/3/0 {
  mtu 2000;
  unit 0 {
    family inet {
      address 192.168.2.1/24;
    }
    family mpls;
  }
}
fxp0 {
  unit 0 {
    family inet {
      address 10.10.10.2/24;
    }
  }
}
lo0 {
  unit 0 {
    family inet {
      address 1.1.1.2/32;
    }
  }
}
```

```

    }
  }
}
}

```

GMPLS 동작과 관련된 routing options 을 위해 특별히 config 해주어야 할 사항은 따로 존재하지 않으며 아래 명령을 이용하여 routing options을 점검해 보기로 한다. 인터넷 접속을 위한 next-hop을 게이트웨이 address인 203.230.116.193을 할당하였다.

```

kisti@MX480> show configuration routing-options
routing-options {
  static {
    route 0.0.0.0/0 next-hop 203.230.116.193;
  }
  router-id 1.1.1.2;
}

```

다음으로 GMPLS 동작과 관련된 protocols의 config 사항을 점검해 보기로 한다. 역시 MPLS 프로토콜에서는 기본적으로 in-band 시그널링을 사용하기 때문에 rsvp 프로토콜 구현을 위해 인터페이스는 데이터 플레인과 동일한 인터페이스를 사용해야 한다. 따라서 rsvp config에서 시그널링을 위한 인터페이스로 xe-4/2/0 과 xe-4/3/0 를 사용하였다. transit 라우터로만 동작하는 MX480 라우터에서는 양방향 MPLS config를 위해 특별히 설정해 줄 사항은 없고 연동을 위해 사용되어지는 인터페이스인 xe-4/2/0, xe-4/3/0 만 mpls {} 안에 설정해 주면 된다. 마지막으로 다른 연동 라우터와의 ospf 프로토콜 동작을 위해 MX480 라우터의 xe-4/2/0, xe-4/3/0 인터페이스를 ospf {} 안에 추가해 주어야 한다. 아래 명령을 사용하여 GMPLS protocol config와 관련된 사항을 점검해 보기로 한다.

```

kisti@MX480> show configuration protocols
protocols {
  rsvp {
    interface xe-4/2/0.0;
    interface xe-4/3/0.0;
    interface lo0.0;
  }
  mpls {
    traffic-engineering bgp-igp-both-ribs;
    interface xe-4/2/0.0;
    interface xe-4/3/0.0;
  }
  ospf {
    area 0.0.0.0 {

```

```

        interface lo0.0 {
            passive;
        }
        interface xe-4/2/0.0;
        interface xe-4/3/0.0;
    }
}
}

```

다음으로 GMPLS config를 통해 실제 lsp가 생성되었는지를 확인하기로 한다. 아래 명령을 사용하면 Ingress 라우터로서, 그리고 Egress 라우터로서, 그리고 Trnsnit 라우터로서 생성된 lsp 세션의 정보를 보여준다. MX480 라우터는 transit 라우터로서만 동작하기 때문에 2개의 Transit sessions lsp (T320-MX240 lsp transit session 1개, MX240-T320 lsp transit session 1개) 만 확인할 수 있다.

```

kisti@MX480> show mpls lsp
Ingress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

```

```

Egress LSP: 0 sessions
Total 0 displayed, Up 0, Down 0

```

```

Transit LSP: 2 sessions

```

To	From	State	Rt	Style	Labelin	Labelout	LSPname
1.1.1.1	1.1.1.3	Up	1	1 FF	299808	3	1:5979:1::9653 :3::1658
1.1.1.3	1.1.1.1	Up	1	1 FF	299792	3	1:5979:1::9653 :3::1658

```

Total 2 displayed, Up 2, Down 0

```

3.3.4 MX240 라우터 configuration 사항 및 테스트

주니퍼 MX240 라우터의 ospf 라우팅 테이블이 제대로 올라가 있는지 확인하였다. GMPLS의 동작 전에 ospf 라우팅 프로토콜을 이용하여 라우터와 라우터 그리고 클러스터 간에 LSA (link state advertisement) 메시지를 이용하여 전체 네트워크의 link state를 확인하기 때문이다. MX240 라우터에서 확인한 ospf 라우팅 테이블의 내용은 다음과 같았다. MX240 라우터 자신의 인터페이스에 연동된 라우터 및 클러스터를 제외한 모든 라우터 및 클러스터의 ip address (192.168.1.0, 192.168.3.0, 192.168.125.0, 192.168.175.0)와 연동되어진 모든 라우터의 루프백 어드레스 (1.1.1.1, 1.1.1.2, 1.1.1.4)를 확인할 수 있었다. 이를 이용하여 각 라우터는 네트워크의 전체 토폴로지 및 상태 정보를 미리 파악할 수 있

다. 아래 명령어를 활용하여 MX240 라우터의 ospf 라우팅 테이블을 확인하였다.

```
kisti@MX240> show route protocol ospf
inet.0: 19 destinations, 24 routes (19 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

1.1.1.1/32      [OSPF/10] 1w2d 15:20:42, metric 2
                > to 192.168.2.1 via xe-2/2/0.0
1.1.1.2/32      *[OSPF/10] 1w2d 15:20:42, metric 1
                > to 192.168.2.1 via xe-2/2/0.0
1.1.1.4/32      *[OSPF/10] 1w2d 15:20:42, metric 3
                > to 192.168.2.1 via xe-2/2/0.0
192.168.1.0/24  *[OSPF/10] 1w2d 15:20:42, metric 2
                > to 192.168.2.1 via xe-2/2/0.0
192.168.3.0/24  *[OSPF/10] 1w2d 15:20:42, metric 3
                > to 192.168.2.1 via xe-2/2/0.0
192.168.125.0/24 *[OSPF/10] 1w2d 15:20:42, metric 3
                > to 192.168.2.1 via xe-2/2/0.0
192.168.175.0/24 *[OSPF/10] 1w0d 00:11:26, metric 4
                > to 192.168.2.1 via xe-2/2/0.0
203.230.116.192/26 [OSPF/10] 1w2d 15:20:42, metric 3
                > to 192.168.2.1 via xe-2/2/0.0
224.0.0.5/32    *[OSPF/10] 1w2d 15:22:27, metric 1
                MultiRecv

inet.3: 2 destinations, 2 routes (2 active, 0 holddown, 0 hidden)

mpls.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
```

다음으로 MX240 라우터의 인터페이스 사항을 다음과 같이 설정하였다. 먼저 MX240 라우터의 ge-2/0/0 인터페이스에 외부접속 및 관리를 위한 ip address로 203.230.116.196 번을 할당하였고 xe-2/2/0 인터페이스는 MX480 라우터와의 연동을 위한 사설 ip address로 192.168.2.2 번을 할당하였다. 키스티 대전 ANTIlab의 클러스터 연동을 위해 xe-2/3/0 인터페이스에 192.168.150.2 번을 할당하였고 MX480 라우터의 루프백 어드레스 (router id)로 1.1.1.3 번을 할당하였다.

```
kisti@MX240> show configuration interfaces
interfaces {
    ge-2/0/0 {
```



```

    unit 0 {
        family inet {
            address 203.230.116.196/26;
        }
    }
}
ge-2/1/0 {
    unit 0 {
        family inet;
    }
}
xe-2/2/0 {
    mtu 2000;
    unit 0 {
        family inet {
            address 192.168.2.2/24;
        }
        family mpls;
    }
}
xe-2/3/0 {
    unit 0 {
        family inet {
            address 192.168.150.2/24;
        }
        family mpls;
    }
}
fxp0 {
    unit 0 {
        family inet {
            address 10.10.10.3/24;
        }
    }
}
lo0 {
    unit 0 {
        family inet {
            address 1.1.1.3/32;
        }
    }
}

```

```
    }  
}
```

다음은 routing options과 관련된 config 설정이다. 역시 인터넷 접속을 위해 next-hop을 게이트웨이 address인 203.230.116.193 으로 할당하였다.

```
kisti@MX240> show configuration routing-options  
routing-options {  
    static {  
        route 0.0.0.0/0 next-hop 203.230.116.193;  
    }  
    router-id 1.1.1.3;  
}
```

다음으로 GMPLS 동작과 관련된 protocols의 config 사항을 점검해 보기로 한다. 역시 MPLS 프로토콜을 사용하여 양방향으로 LSP와 대역폭을 보장해주는 방식으로 GMPLS 프로토콜을 구현 하였다. 역시 rsvp 시그널링 구현을 위해 데이터 플레인과 동일한 인터페이스인 xe-2/2/0를 rsvp {} 안에 정의하였다.

대전 통신실에 위치한 T320 라우터와의 GMPLS config를 위해 MX240 라우터에서는 단방향 MPLS config 설정을 필요로 한다. T320 라우터에 연결된 클러스터까지 FEC 호스트 기반의 설정을 위해 install 192.168.125.7 명령을 필요로 하며, 대역폭 설정을 위해 bandwidth 5m 라는 명령어를 활용하여 5M 대역폭을 보장했으며, label-switched-path 경로는 1:5979:1::9653:3::1658 이라는 NRM에 의해 전달받은 GRR ID를 사용하였다. 마지막으로 다른 연동 라우터와의 ospf 프로토콜 동작을 위해 MX240 라우터의 ge-0/0/0, xe-2/2/0, xe-2/3/0 인터페이스를 ospf {} 안에 추가해 주었다. 아래 명령을 사용하여 protocol config와 관련된 사항을 점검해 보기로 한다.

```
kisti@MX240> show configuration protocols  
protocols {  
    rsvp {  
        interface xe-2/2/0.0;  
        interface lo0.0;  
    }  
    mpls {  
        traffic-engineering bgp-igp-both-ribs;  
        label-switched-path 1:5979:1::9653:3::1658 {  
            to 1.1.1.1;  
            install 192.168.125.7/32;  
            lsp-attributes {
```

```

        switching-type psc-1;
        encoding-type packet;
    }
    bandwidth 5m;
    no-cspf;
    primary 1:5979:1::9653:3::1658;
}
path 1:5979:1::9653:3::1658 {
    1.1.1.1 loose;
}
interface lo0.0;
interface xe-2/2/0.0;
}

ospf {
    area 0.0.0.0 {
        interface lo0.0 {
            passive;
        }
        interface xe-2/2/0.0;
        interface xe-2/3/0.0;
    }
}
}

```

다음으로 GMPLS config를 통해 실제 lsp가 생성되었는지를 확인하기로 한다. 아래 명령을 사용하면 Ingress 라우터로서, 그리고 Egress 라우터로서, 그리고 Trnsnit 라우터로서 생성된 lsp 세션의 정보를 보여준다. 첫 번째 결과가 앞서 config를 통해 생성된 Ingress lsp 세션이며, 두 번째 Egress lsp 세션의 결과는 상대방 라우터 즉 T320 라우터에서 MX480 라우터 쪽으로의 lsp 설정을 위해 T320 라우터에서 MPLS config한 것에 대한 결과이다. 마지막으로 transit 라우터로서 lsp 세션은 없음을 확인할 수 있다.

```
kisti@MX240> show mpls lsp
```

```
Ingress LSP: 1 sessions
```

To	From	State	Rt P	ActivePath	LSPname
1.1.1.1	1.1.1.3	Up	3*	1:5979:1::9653:3::1658	1:5979:1:: 9653:3::1658

Total 1 displayed, Up 1, Down 0

```
Egress LSP: 1 sessions
```

To	From	State	Rt Style	Labelin Labelout	LSPname
----	------	-------	----------	------------------	---------

```
1.1.1.3      1.1.1.1    Up    0 1 FF      3          - 1:5979:1::9653:3::1658
Total 1 displayed, Up 1, Down 0
```

```
Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
```

마지막으로 ip 라우팅이 아닌 실제 설정된 lsp 경로를 통해 패킷이 label swapping을 통해 전송되는지를 확인하였다. ping 192.168.125.7 rapid count 1000 명령어를 사용하여 T320 라우터에 연결된 클러스터 까지 1000개의 패킷을 전송해 보았다.

```
kisti@MX240> ping 192.168.125.7 rapid count 1000
PING 192.168.125.7 (192.168.125.7): 56 data bytes
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
--- 192.168.125.7 ping statistics ---
1000 packets transmitted, 1000 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.277/0.363/37.026/1.569 ms
```

lsp를 통해 실제 패킷이 전송되는 것을 확인해 보기 위해서는 show mpls lsp statistics 명령을 이용하면 된다. 아래 결과처럼 1000개의 패킷이 lsp 경로를 통해 MX240 라우터에 연결된 클러스터까지 전송됨을 확인하였다.

```
kisti@MX240> show mpls lsp statistics
Ingress LSP: 1 sessions
To          From          State  Packets  Bytes LSPname
1.1.1.1     1.1.1.3      Up     1000     84099 1:5979:1::965
3:3::1658
Total 1 displayed, Up 1, Down 0
```

Egress LSP: 1 sessions

To	From	State	Packets	Bytes	LSPname
1.1.1.3	1.1.1.1	Up	NA	NA	1:5979:1::965
3:3::1658					

Total 1 displayed, Up 1, Down 0

Transit LSP: 0 sessions

Total 0 displayed, Up 0, Down 0

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 및 컴퓨팅 자원을 예약 할당해 줄 수 있는 통합자원관리시스템 기술이 필수적이라 말 할 수 있다. Heterogeneous한 네트워크 환경에서 동적으로 연결 지향형 네트워크를 구성하기 위해 GMPLS 제어평면 기술은 가장 보편적인 기술이라 할 수 있다. 본 보고서에서는 연구망에 분산되어 있는 컴퓨팅 자원 및 네트워크 자원들에 대해 개발된 통합자원관리시스템을 활용하여 기본적인 자원 예약 및 할당 시험을 할 수 있도록 유니퍼사의 MX240, MX480, T640, T320 라우터를 활용하여 개발되어진 GMPLS 테스트베드 망에 대하여 기술하였다. GMPLS는 다계층 네트워크를 관통하여 하나의 LSP를 만들어 상이한 네트워크 기술 간에 통합된 연결 제어 메커니즘을 제공한다. 이것은 안정적이고 보장된 대역폭을 보장받기를 원하는 e-Science 등 융합 응용들의 협업 연구에 좀 더 유연한 네트워크 환경을 제공한다. GMPLS 망 상에서 융합망 자원의 관리 및 할당을 위한 통합자원관리 시스템과의 연동을 위해, 주니퍼 네트워크 장비인 T640, T320, MX480, MX240 라우터 장비를 활용하여 3계층 GMPLS 테스트베드 망을 개발하여 보고 구현된 GMPLS 테스트베드망 상에서 융합망 통한자원관리 시스템을 통해 각 시스템 (GRS, NRM, CRM) 의 기본 기능을 검증 하였다. 사용되어진 각 T320, MX480, MX240 라우터 장비는 유니퍼에서 제공되어진 JUNOS 9.1R1.8 version OS가 탑재되었다. 구현된 GMPLS 테스트베드망 상에서 e-Science와 같은 융합망 어플리케이션의 컴퓨팅, 및 네트워크 자원의 관리 및 예약, 할당을 위해 통합자원관리시스템을 개발 적용하였다. 현재까지 개발된 통합자원관리시스템을 살펴보면 다음과 같다. 먼저 GRS 작업(컴퓨팅/네트워크 자원을 동시에 활용) 제출을 위한 웹기반 사용자 인터페이스를 개발하였으며, GRS 작업 모니터링을 위한 웹기반 사용자 인터페이스를 개발하였다. 또한 국제 표준 규격 GNSI 기반의 웹서비스를 지원하는 NRM을 개발하였으며, 타임 테이블 기반의 컴퓨팅 자원 관리 시스템(CRM)을 개발 완료하였다. 현재까지 개발된 시스템들은 관리자모드의 시범서비스용이며 2011년 2012년 개발 안전화 단계를 거쳐 실제 사용자에게 서비스가 가능한 시스템으로 전환을 할 예정이다. 개발되어진 NRM(네트워크 자원 관리 시스템), CRM(컴퓨팅 자원 관리 시스템), GRS(글로벌 스케줄러)과 서울-대전 사이에 구현된 GMPLS 테스트베드망 및 클러스터 자원과의 연동을 통해 각 시스템의 기본 기능을 테스트하였고 가장 기본적인 자원예약 및 할당 시나리오에 따라 자원 할당 시연이 가능하였다.

주니퍼 라우터의 GMPLS 테스트베드망 개발을 위한 구성도는 다음과 같았다. 클러스터는 서울 키스티 통신실의 T640 라우터에, 그리고 대전 키스티 통신실의 T320 라우터에 그리고 대전 키스티 ANTIlab의 MX240 라우터에 각각 연동되어 있으며, 컴퓨팅 자원을 위한 예약으로 사용되어 진다. 각 라우터에 연동을 위한 configuration을 위해 외부접속 공인 IP를 기가비트 인터페이스에 하나씩 할당하였으며 각 라우터의 GMPLS 데이터평면 연동을 위해서는 사설 IP인 192.168.X.X 대역을 할당하였다. GRS는 GMPLS 테스트베드망 위에 외부접속 공인 IP를 통해 각각 CRM (컴퓨팅 자원관리시스템), NRM (네트워크 자원관리시스

템)의 외부접속 공인 IP와 서로 통신하며 통합자원관리 역할을 수행한다. 그리고 GMPLS config의 하나의 예로서 T640 라우터에 연결되어있는 클러스터와 T320라우터에 연결되어 있는 클러스터, 그리고 T320라우터에 연결되어 있는 클러스터와 MX240에 연결되어 있는 클러스터를 각각 FEC 호스트기반으로 config 하였다.

참고문헌

- [1] "MPLS Applications Configuration Guide", Release 10.0, Juniper Networks, Inc., Oct., 2009.
- [2] 2008년, 2009년 과학기술자원 융합망 일반사업보고서, 한국과학기술정보연구원
- [3] GMPLS 기반의 광경로 할당 알고리즘 개발 및 구현," 한국과학기술정보연구원 2009년 위탁연구과제보고서, 안동대학교
- [4] 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.
- [5] 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
- [6] 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
- [7] 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
- [8] Xi Yang, Tom Lehman and Chris Tracy, "Policy-based Resource Management and Service Provisioning in GMPLS Networks," INFOCOM 2006.
- [9] Erik-Jan Bos, Gigi Karmous-Edwards, "GLIF Universal Service Interface GUSI A GLIF Project," GLIF 2008.
- [10] G-Lambda, "Inter-domain advance reservation of coordinated network and computing resources over the Pacific," GLIF 2007
- [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] The G-lambda project, "Grid Network Service-Web Services Interface, version 3," OGF, 2008