GMPLS Tutorial and R&E Network Implementation

An overview of the GMPLS framework, protocol descriptions, open issues, implementations, research Testbeds and OnRamp to GMPLS R&E Networks

Wes Doonan, John Moore, Jerry Sobieski, Rick Summerhill, Chris Tracy

Albuquerque, NM
February 5, 2006
Just checking…

• Top level bullet
  – Next level bullet
    • Bullet level 3

• Can you hear?
  – Check 1…2…3…Check

• Too hot?
  – Too cold?
Introduction

• Who are we?

• Why do we think this is (more) important (than the SuperBowl)?
Tutorial Contents

- GMPLS Control Plane Overview
  - Fundamentals, MPLS Protocols, GMPLS Extensions, Standards, Interoperability, Notes
- GMPLS Implementations
  - Application to R&E Community
  - Research Testbeds Overview
  - DRAGON/HOPI Architecture
- OnRamp to GMPLS R&E Networks
  - Deploying your own
  - DRAGON Software Tutorial and Demo
  - Connecting to HOPI
Where are we?

• **GMPLS Control Plane Overview**
  – Fundamentals, MPLS Protocols, GMPLS Extensions, Standards, Interoperability, Notes

• **GMPLS Implementations**
  – Application to R&E Community
  – Research Testbeds Overview
  – DRAGON/HOPI Architecture

• **OnRamp to GMPLS R&E Networks**
  – Deploying your own
  – DRAGON Software Tutorial and Demo
  – Connecting to HOPI
GMPLS
Control Plane Overview

• Introducing Wes Doonan
  – Movaz Networks
Control Plane Overview

Wes Doonan
wes@movaz.com
Agenda

- Fundamentals
- MPLS Protocols
- GMPLS Extensions
- Standards and Interoperability
- Application Notes
MPLS and GMPLS manage data paths within transport networks using IP based control mechanisms.

Data paths are identified using a label.

MPLS was conceived as a packet-based technology:
- The payload originally focused on IP.
- Forwarding is packet or cell-based:
  - Each packet, or cell, is labeled.
  - Forwarding involves label look-up and swap.

GMPLS adds support for:
- SONET/SDH (TDM), Optical (Lambdas), and Ports.
- Includes packet support.

Provide a foundation for data transport applications.
MPLS Labels

- Labels are identifiers used in forwarding
- Label values have meaning between two nodes NOT end-to-end
- There may be multiple labels stacked per packet
- Format:

```
<table>
<thead>
<tr>
<th>Label Stack Entry Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
</tr>
<tr>
<td>---</td>
</tr>
</tbody>
</table>
```

- Label: Label Value, 20 bits (0-16 reserved)
- Exp.: Experimental, 3 bits (Class of Service is one use)
- S: Bottom of Stack, 1 bit (1 = last entry in label stack)
- TTL: Time to Live, 8 bits
Traditional MPLS Basics

- MPLS Domains
- MPLS Nodes and Label Switch Routers

- Label Switch Path
- Label Edge Routers
  - Ingress, egress and transit nodes
LSP Types

- **Prefix based**
  - LSPs created based on “route” advertisement
  - Controlled by routing or signaling
  - Some application at the edge

- **Tunnel based**
  - LSPs created between specific MPLS end-points
  - Controlled by signaling
  - Applicable to the core

- **MPLS LSPs are always unidirectional**
  - Two LSPs required to support bi-directional traffic
  - GMPLS adds bi-directional support

- Different control protocol supports different types
MPLS Prefix LSP: Example

- Labels distributed based on routes

- Downstream LSRs learn routes
  - Downstream LSRs advertise mappings
  - Mappings are propagated upstream
  - Upstream LSRs may also request mappings
MPLS Tunnel LSP: Example

- Labels distributed between specific end-points

- Ingress initiates LSP
  - Request propagated to egress
  - Egress responds with label
  - Response propagated upstream to ingress

Data flow
GMPLS Controlled Lightpath

- Reuse of MPLS and IP Control

- Ingress initiates lightpath setup
- Request propagated to egress
- Egress responds with lambda
- Response propagated upstream to ingress
There are multiple MPLS signaling protocols
- LDP vs. CR-LDP vs RSVP-TE
- RSVP-TE is widely deployed, CR-LDP development ceased in IETF WGs

Reuses IP routing protocols
- OSPF-TE, ISIS-TE – both deployed
- BGP (VPNs)

History:
GMPLS History

- One real signaling protocol
  - GMPLS-RSVP

- One routing protocol
  - GMPLS-OSPF is predominant in optical space
    GMPLS-ISIS will be used in other markets

- Link Management Protocol (LMP)
  - Used to identify and test data paths independent from control

- History:

  - Cisco, WCOM
  - MPλS
  - IETF MPLS/CCAMP Working Group
  - SONET/SDH Ports, … (many)
  - GMPLS
  - GMPLS-RSVP
  - GMPLS-ISIS
  - GMPLS-OSPF
  - LMP
  - NTIP
  - GMPLS-LDP

Copyright Movaz Networks, Inc. © 2006
MPLS and GMPLS Functions

Signaling: Dynamic control of service delivery

Path Computation: Selects TE / CR paths within a network (With relevant constraints, aka CSPF)

Routing: Provides topology and resource availability to all nodes

Link Management: Identifies inter-node data connection properties
MPLS and GMPLS Protocols

**MPLS Control Plane**
- Prefix LSP Signaling
- LDP (and BGP)
- Tunnel LSP Signaling
  - RSVP-TE, CR-LDP
- TE-Routing
  - OSPF-TE, ISIS-TE

**GMPLS extensions**
- GMPLS-Signaling
- RSVP-TE
- GMPLS-Routing
  - OSPF-TE, ISIS-TE
- Link Management
  - LMP, LMP-WDM, LMP-SONET
Prefix LSP Signaling - LDP

- **LDP**
  - General LDP information
  - LDP basics
  - Modes of label distribution and management

- **Supports**
  - IP prefix/route based label establishment

- **Does not support**
  - Multicast
  - QoS, CoS
  - Multi-path
Phases of LDP Operation

- Peer Discovery
- Transport connection establishment
- Session initialization
- Label exchange
- Session maintenance
Label Distribution

- Upstream nodes send Label Request Messages
  - Includes one or more FECs (hosts/prefixes)

- Downstream nodes send *Label Mapping Messages*
  - Provides FEC to Label mappings

- The ordering of each message is governed by label distribution mode
Tunnel LSP Signaling

- RSVP-TE
  - General RSVP information
  - RSVP basics
  - RSVP-TE and related extensions

- CR-LDP
  - Parallels RSVP-TE – omitted
RSVP MPLS Support

- RSVP-TE aka RSVP-Tunnel
  - Standardized by IETF MPLS working group
  - Adds tunnel LSP control and other MPLS related features to standard RSVP

- RSVP Scaling Extensions
  - Standardized by IETF RSVP working group
  - Are not MPLS specific
  - Reduces per session (LSP) and message overhead
  - Adds message reliability
RSVP and Int-Serv Background

- Are IETF Standard RFCs (1997)
- RSVP provides QoS signaling for IP
  - Does not support LSP establishment
  - Relies on routing for next hop and loop avoidance
- Int Serv provides QoS definitions used by RSVP
  - Guaranteed
    - Approximates a fixed size link
    - Guarantees both delay and bandwidth
  - Controlled load
    - Approximates a lightly loaded network
    - Provides low loss and low latency variance
RSVP Basics

- RSVP provides a way to make resource allocation requests for specific data flows
  - Standards support IPv4 and IPv6 TCP, UDP, and IPSec flows
- Targeted more at the edge of the network
- Considers multicast first, then unicast
  - Discussion will be limited to MPLS relevant features
- Assumes IP header on data packets
- Supports shared and per sender allocations
- RSVP messages are sent directly over IP
Phases of RSVP Operation

- Source sends Path message
- Path propagated to receiver
- Receiver sends Resv Message
- Previous hop allocates resources
- Path message propagated to source
  - Allocations made along the way
- Data path allocated
- Data path allocated
- Path and Resv state refreshed
LSRs establish tunnels using standard RSVP procedures

Path message includes new label related objects
- Identifies session as MPLS, requests label and identifies sender

Resv message includes label

Data traffic must be sent with label passed in Resv
- Encapsulated data may be of any type

Only unicast tunnels, and FF and SE styles are supported
- SE used in make before break
RSVP-TE Extensions

- Builds on RSVP’s formats and mechanisms
  - Including state machines and implementations
- Adds support for:
  - LSP tunnel establishment and labeled flows
    - Supports merging and make-before-break
  - Explicit and record route functions
    - For traffic engineering
  - Preemption
  - Neighbor failure detection
Tunnel Applications

- Traffic Engineering (TE)
- IP Service
  - QoS provisioned
  - DiffServ, SLAs
- VPN / VPLS Service
- Foundation for GMPLS
# MPLS Signaling Protocol Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Protocol</th>
<th>LDP</th>
<th>CR-LDP</th>
<th>RSVP-TE</th>
<th>RSVP-Ext</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix LSPs</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tunnel LSPs</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>QoS</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Explicit Route</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Record Route</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Route Pinning</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Preemption</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Rapid Failure Detection</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Reliable Delivery</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Message Aggregation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>LSP Scaling</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
MPLS Routing

- OSPF and ISIS-TE Extensions
  - Provide equivalent functionality
  - Provides information for path selection
    - Aka Constrained SPF / CSPF
  - Add TE Opaque LSAs
    - TE Router LSA – Identifies TE router
    - TE Link LSA
Routing - TE Link Information

- TE Link LSA
  - Link type
    - Point-to-point or multi-point
  - Link ID
    - Neighbor router or DR ID
  - Local interface IP address
  - Remote interface IP address

- Traffic engineering metric
  - Maximum bandwidth
  - Maximum reservable bandwidth
  - Unreserved bandwidth
  - Administrative group
    - Also called Resource Class/Color
Generalized MPLS

Automation tool for circuit provisioning

- GMPLS is a superset of MPLS and MP\(\lambda\)S
- All about IP control protocols reuse

**GMPLS Control Plane**
RSVP-TE, CR-LDP
OSPF, IS-IS

**Data Plane**
MPLS, SONET/SDH,
\(\lambda\)s/WDM, Ports
"label" == shim header
"label" == TDM time slot
Lambda Switching

“label” == lambda frequency

\{\lambda\ldots\lambda\}_1 \quad \lambda_1 \quad \lambda \quad \{\lambda\ldots\lambda\}_1

\{\lambda\ldots\lambda\}_{10} \quad \lambda_{100} \quad \lambda_p - \lambda_1 \quad \{\lambda\ldots\lambda\}_{10}
Port Switching

“label” == fiber port
MPLS labels are a simple 32 bit quantities

GMPLS extends these to arbitrary values to cover all switching technologies

- MPLS labels
- Timeslots
- Wavelengths, wavebands and fibers

Label type can be deduced from the link type or requested in the Generalized Label Request
Some switches cannot modify labels (lambdas)
May want to restrict choice based on available resources
Advertise available lambdas using ‘Label Set’
Label set is allowed to have just one member
Some optical switches are slow to program
‘Suggested Label’ shows choice at upstream
Allows signaling to be pipelined
May require reprogramming later
### Bidirectional LSPs

- **MPLS approach**
  - Manual configuration (SNMP, Telnet, etc.)
  - Inferred
  - Management and signaling overhead

- **GMPLS uses ‘Upstream Label’**
  - Single signaling exchange
  - Resources provisioned during forwards signaling
  - Error response may indicate available labels
First node imposes red using Upstream Label
Second node attempts to impose red
Third node rejects red, but offers yellow, blue and green in ‘Acceptable Label Set’
Second node tries again using yellow
Third node imposes yellow
Enhanced Error Reporting

- New flag on PathErr to show state removal
- Non-adjacent error notification
Control & Data Plane Separation

- Separate control and data channels
  - Out of band in fiber, out of band out of fiber
  - Data and control can fail independently
  - Control channel failure does not disrupt data
  - Control plane reboot does not disturb data plane
Alarm Free Control

- Supported via Administrative status
- Administrative status carried on
  - Signaling requests and responses
- Alarm-free setup
- Alarm-free teardown
- Alarm status control during protection switchover
- Can place LSP into ‘test’ status
Other Extensions

- Hierarchy
- Bundling
- Unnumbered
- Fast re-route
- Overlay
- Crankback
- OAM

- Not to mention MPLS/GMPLS based applications!
Switching Hierarchies

- TDM link (e.g. SONET)
- Individual lambda
- Whole fiber switching
- Packet
Overlay Model

- Separate and independent Control Planes
- Service Request interface can be
  - Mediation entity such as OSS or CNM
  - Management layer (proprietary or SNMP)
  - Standard GMPLS
  - User to Network Interface (UNI)
Peer Model

- Full information across technologies
- Same routing and signaling protocols at peering points
- Different granularity of bandwidth handled through “tunnels” or “virtual links” requested by traffic grooming points
Hybrid Model

- Peer model “leaks” information between layers in the vertically integrated network
  - Peer model will remain unpopular for use between Service Providers
- Overlay is “simple” but restricts service delivery
- Augmented model limits information exchange
  - Controlled exchange of reachability
  - Allows single or multiple signaling protocols
  - Facilitates Optical VPNs (L1 VPNs)
  - Parallels how the Internet is built
GMPLS Routing Extensions

- Extensions to OSPF and IS-IS to supply required information in opaque LSAs
- Already have TE extensions for MPLS:
  - Resource availability
- Further additions for GMPLS:
  - Unnumbered interface support
  - Link protection type (unprotected, shared, 1:1, 1+1, enhanced)
  - Shared Risk Link Group (SRLG)
  - Interface Switching Capability:
    - Identifies type of switching supported and Maximum LSP bandwidth per priority
  - Graceful restart considerations
Link Management

- Need to address a common set of issues
  - Isolation of faults transparent networks
  - Scale the number of links without increasing configuration
  - Scale the parallel (‘bundled’) links without increasing the amount of information advertised by the IGP
    - Handle case where transmission (e.g., WDM) and switching are separated = very large number of component / per-lambda links
- New protocol needed to resolve these issues, e.g. LMP
LMP

- Control Channel Management
  - Maintain an IP control channel between LMP peers
- Link Property Correlation
  - Discover and agree data link properties
- Link Verification
  - Map interface IDs and verify data connectivity
- Fault Management
  - Detect and isolate faults
- Authentication
LMP Fault Localization

- **Nodes detecting failure** send message upstream
- **Upstream node**
  - May know about error and have sent message
  - Can test to see if it has error on upstream link
  - Responds and reports error locally
  - Needed when information not available from hardware
MPLS MIBs already exist for
- Modeling the cross-connects in an LSR
- Requesting and controlling TE-LSPs

Enhancements are being made to
- Support wider definition of ‘label’
- Allow control of new GMPLS features

Other new MIBs for
- LMP
- Link bundling
GMPLS Status

1. Routing
   1. RFC2328 (OSPF v2)
   2. RFC3630 (OSPF-TE v2)
   3. RFC4202 (GMPLS Routing)
   4. RFC4203 (OSPF-GMPLS)

2. Signaling
   1. RFC2205 (RSVP)
   2. RFC3209 (RSVP-TE)
   3. RFC3471 (GMPLS Signaling)
   4. RFC3473 (RSVP-GMPLS)

3. In Progress
   1. draft-ietf-ospf-ospfv3-traffic-06
   2. draft-ietf-ccamp-rsvp-restart-ext-05
   3. draft-ietf-ccamp-crankback-05
   4. draft-ietf-ccamp-gmpls-segment-recovery-02
   5. draft-ietf-ccamp-gmpls-recovery-e2e-signaling-03
   6. draft-ietf-ccamp-gmpls-alarm-spec-03
   7. draft-ietf-ccamp-gmpls-addressing-02

Mature, several interoperable implementations
GMPLS Major Works In Progress

- GMPLS Protection / Restoration
- Multi-area
  - Inter-provider, inter-area, etc.,
- ASON support / coordination
- Constraints and impairments for all-optical networks
- Point-to-Multipoint
Interoperability (ISOCORE)

Navtel
Juniper
Movaz
Sycamore
Avici
Tellabs
Completed 10/03
Report available to ISOCORE members

= OC48
= OC12
= GbE
Interoperability (UNH)

- Fast Ether as Control LINK
- OC48 as TE-LINK
- OC48 or others as TE-LINK
- Lambda as TE-LINK
- N*OC48 or OC-192? as TE-LINK

Completed 01/04
Report available
The Ideal

RON = Regional Optical Network
ION = Intercontinental Optical Network

"Request 5Gps"

"OK!"

"Receiving 5Gps"

- = Packet Link
- = Lambda Link
- = GMPLS LSP

RON = Regional Optical Network
ION = Intercontinental Optical Network
The Reality

How to cross the control plane gap?

“OK!”

“Uh, well …”

“Request 5Gps”
Option 1: Best Effort

1. GMPLS LSP from U1 to N1
2. GMPLS LSP from N2 to U2

- Each segment provisioned separately
- Best-effort in the middle
- IP network uninvolved in control
Option 2: Manual Stitching

1. GMPLS LSP from U1 to N1
2. MPLS LSP from N1 to N2
3. GMPLS LSP from N2 to U2

- Each segment provisioned separately
- Some QoS in the middle
- IP network must be MPLS-enabled
Option 3: End-to-End MPLS

1. GMPLS LSP from U1 to N1, forms FA
2. GMPLS LSP from N2 to U2, forms FA
3. MPLS LSP from U1 to U2, thru FAs

- MPLS is signaled end-to-end
- Routers at RONs must support FA-LSPs
- IP network must be MPLS-enabled, share signaling with RONs
Option 4: Tunneled GMPLS

1. Configured tunnel (GRE, IPIP) + TE link
2. GMPLS LSP from U1 to U2, thru tunnel

- GMPLS is signaled end-to-end
- Best-effort in the middle
- IP network uninvolved in control
Option 5: End-to-End GMPLS

1. GMPLS LSP from U1 to U2

- GMPLS is signaled end-to-end
- IP network must be GMPLS-enabled, share signaling with RONs
- Realizable via UNI/NNI, hierarchy
GMPLS has evolved from MPLS
- Builds on MPLS’ and IP’s operational experience
- GMPLS adds features required for the management of Optical networks
- GMPLS managed services can be integrated across technologies

- GMPLS enabled optical products are available today
Thank You
GMPLS Tutorial and R&E Network Implementation

- Introducing Chris Tracy
  - Mid-Atlantic Crossroads
GMPLS Tutorial and R&E Network Implementation

- **New material!**
  - Being presented for the first time today…

- **Not intended to be the GMPLS bible**
  - But we hope it will help explain issues associated with the deployment of GMPLS…

- **Please share your comments/suggestions**
  - It will help us to improve this presentation in the future
Where are we?

- **GMPLS Control Plane Overview**
  - Fundamentals, MPLS Protocols, GMPLS Extensions, Standards, Interoperability, Notes
- **GMPLS Implementations**
  - Application to R&E Community
  - Generic Control Plane Architecture
  - Case Study: HOPI Control Plane
- **OnRamp to GMPLS R&E Networks**
  - DRAGON Software Tutorial and Demo
  - Connecting to HOPI
GMPLS Implementations

• Application to R&E Community
  – The Emerging Global Application
  – What can GMPLS provide?
  – What is missing?
  – Why should we use GMPLS?

• Generic Control Plane Architecture
  – Overview
  – Generic Network Element (NE)
  – Connection Provisioning
  – Interface Addressing
  – LSP Establishment

• Case Study: HOPI Control Plane
The Emerging Global Application

Emerging large-scale, globally distributed applications require more sophisticated network services than have previously been available:

- **Dedicated network resources**
  - An application needn’t worry about its impact on other network users, or vice versa

- **Deterministic performance**
  - Repeatable and predictable from day to day / year to year

- **Very high performance**…
  - Multi-Gbs flows, low latency/loss, minimal jitter, global reach

- **Reservable and schedulable in advance**
  - Particularly in conjunction with availability of non-network resources (e.g. radio telescopes, computational clusters, etc.)

- **Flexible and dynamic**
  - Able to acquire these dedicated network resources on short notice from many potential service/resource providers
Application to R&E Community

• Demands on the network from large-scale apps:
  - High performance
    • Bulk data transfers (terabytes, petabytes)
    • Typically 10 Gb/s and higher
  - Deterministic performance
    • Low-latency for interactive applications, steering, etc.
    • Typically lower bandwidth requirements
  - Combination of both
    • High-bandwidth visualization flow in one direction
    • Lower-bandwidth steering/control data in the other direction
• On-demand, dedicated resources address these issues
  - GMPLS is a framework to accomplish this
    • Provide the end users/applications with dedicated channels
  - Typical Internet connections have problems meeting these demands
    • $\geq 10\text{Gb/s}$ bandwidths only found in the backbone/core
    • Shared bandwidth lacks stability needed for deterministic performance
Application to R&E Community

• What can we expect GMPLS to provide?
  – Routing
  – Signaling
  – Path Computation
  – Link Management
  – Immediate Response

• What is absent that we really want?
  – Advance Scheduling
  – Inter-domain End-to-end Path Computation
  – Policy-based Resource Management
  – Authentication, Authorization and Accounting
  – Control Plane Security
  – Application Specific Topologies
Application to R&E Community

• Why should we use GMPLS?
  – Lots of standards work already done
    • IETF - GMPLS-RSVP-TE, GMPLS-OSPF-TE, LMP
    • ITU-T - ASON
    • OIF - UNI/NNI
  – Vendor implementations already exist
    • Movaz, Ciena, Nortel, Juniper, Calient, etc.
      – (in various states of development)
  – R&E community should leverage this existing work as much as possible
    • Standards-based
    • Promotes interoperability
    • Lowers operating expenses, risk of inadvertent mistakes and provisioning turn-around time
Generic Control Plane Architecture

- **Optical Network Provisioning**
- **Optical Network Planes**
- **Control Plane Interfaces**
  - UNI, I-NNI, E-NNI
- **Control Plane Abstraction**
  - “proxy” agents
- **Inside a GMPLS Network Element**
- **Connection Provisioning**
  - Centralized Provisioning
  - Signaling-based Provisioning
  - Label Request and Response
- **Interface Addressing**
- **GMPLS LSP Tunnel Establishment /w RSVP-TE**
Optical Network Provisioning

- Consider a case where a link is to be provisioned between 2 geographically separated IP routers
  - Could be provisioned using a SONET path layer connection
  - R1 and R2 are *clients* of the optical network
  - O1 and O2 could be OEO optical switching elements (OXC s)
Optical Network Provisioning

- O1-O2 connection would often include WDM systems and amplifiers, as illustrated below
- SONET path layer connection between R1 and R2 requires *connection provisioning*
  - Connection route within optical network needs to be determined
  - Next, the SONET line layer connection segments must be established between the switches
  - Finally, the switches must be cross-connected properly for the end-to-end connection between R1 and R2 to be established
Optical Network Provisioning

- Older optical systems may require manual provisioning
  - No integrated control intelligence
- Automatic connection provisioning requires some control intelligence in the optical network
  - Intelligence may be centralized
    - Central management system
    - Use a well-defined interface between management system and every network element (NE)
  - Intelligence may be distributed
    - Distributed signaling and routing
    - Central management system could still be used, but doesn’t have to talk to every network element

- Our focus today is on the latter:
  - Distributed control intelligence
  - ...or what is often referred to as the control plane
Optical Network Planes

- **Transport Plane (aka Data Plane)**
  - Logic and hardware required to physically transfer data

- **Control Plane**
  - Distributed intelligence that is *only* concerned with the setup and maintenance of connections (cross-connects)
    - Very focused on a select set of tasks
    - Typically implemented using various protocols:
      - Signaling protocols
      - Routing protocols
      - Discovery protocols

- **Management Plane**
  - Systems and protocols that are used to manage the network and its services
    - Fault management, configuration management, accounting, performance monitoring, security management
    - Often provided as TL1/SNMP/CLI-based interfaces
    - Control plane can drive requirements of the management system
      - E.g., monitor signaling statistics for control plane health
Control Plane Interfaces

- **User-Network Interface (UNI)**
  - Between client node and the provider network
- **Interior Network-Network Interface (I-NNI)**
  - Between two nodes within one control domain
- **Exterior Network-Network Interface (E-NNI)**
  - Between two nodes in different control domains
Control Plane Interfaces

- In IP networks, signaling and routing protocols are usually built directly into routers.
- Not necessarily the case with optical networks!
  - Control plane functions distinct from transport functions.
  - Could be built into NE or be physically separate from NE.
  - Single control entity *could* represent multiple NEs.
Control Plane Adjacency

- Connection between control plane agents below defines a control plane **adjacency**
- Adjacent control plane agents do **not** need direct physical connectivity
  - **Reachability** is all that is required
Inside a GMPLS Network Element

- **What happens in a GMPLS Network Element?**
  - Generic scenario for transit node (not source/destination)
  - Assume Explicit Route Object (ERO) contained in PATH
Case Study:
HOPI Control Plane

- Overview
- Useability
- Control Plane Overview
- Control Plane Components
  - NARB
  - VLSR
The Internet2
Hybrid Optical+Packet Infrastructure

**HOPI**

- HOPI is an experimental testbed to deploy, test, evolve, and evaluate new network technologies and architectures to address the needs of emerging global e-science applications.

- **HOPI resources:**
  - Wavelength transport capabilities from the National Lambda Rail, regional optical networks, and commercial providers
  - High performance IP packet services from the Abilene network, regional R&E networks, and open exchange points such as MANLAN
  - Expertise of the network research and e-science applications communities to provide network architecture, engineering, and middleware development, integration and validation.
HOPI Useability

• **Enable a wide base of users to connect and experiment with the facility**
  - Establish a Testbed Support Center to assist with operations, systems development and testing, engineering and applications
    - Mid-Atlantic Crossroads, North Carolina Research and Education Network, and Indiana University GRNOC
  - Integrate Abilene reach and performance capabilities
  - Incorporate international links and exchange points
    - MANLAN, StarLight, ...

• **Dynamic – Learn how to allocate dedicated resources by user request**
  - Deploy a low level control plane foundation for developing automated middleware to allocate, reserve, and provision higher layer networks services

• **Open and [where available] standards based:**
  - Open source DRAGON Software Suite:
  - GMPLS protocols for routing, signaling: GMPLS-OSPF-TE and GMPLS-RSVP-TE to control the switching elements (VLSR)
  - Inter-domain Service Routing (NARB)
    - End-to-End Path Computation
    - AAA
    - Advanced scheduling/reservations
The HOPI Node Architecture

10GE East
10GE West

NLR

Abilene

Hewlett Packard

Performance PC
Measurement PC
Control PC

Force10 E600 Ethernet Switch

Glimmerglass Fiber Switch

Regional Connectors

Other Switching Hardware (TBD)
The HOPI Control Plane

- **DRAGON Software Suite** provides the dynamic routing and provisioning capability in HOPI:
  - **Virtual Label Swapping Router (VLSR)**
    - Open source implementations of GMPLS-OSPF-TE and GMPLS-RSVP-TE protocols
    - Protocols run under FreeBSD or Linux
    - The VLSR translates network protocol events into SNMP/TL1/CLI transactions to reconfigure the switching elements
  - **Network Aware Resource Broker (NARB)**
    - Domain specific agent that listens to OSPF for internal link state changes
    - Provides inter-domain service capability announcements and topology summarization
    - Creates loose hop ERO for RSVP ‘PATH’ requests across multiple domains
    - Performs request authorization and book ahead reservations
Case Study: HOPI Control Plane

Virtual Label Switching Router (VLSR)
- Open source protocols running on PC act as GMPLS network element
- Control PCs participate in protocol exchanges and provisions covered switch according to protocol events (PATH setup, PATH tear down, state query, etc)
Virtual Label Switched Router: VLSR

- Many networks consist of switching components that do not speak GMPLS, e.g. current ethernet switches, fiber switches, etc.
- Contiguous sets of such components can be abstracted into a “Virtual Label Switched Router”
- The VLSR implements Open Source versions of GMPLS-OSPF-TE and GMPLS-RSVP-TE and runs on a Unix based PC/workstation
  - Zebra OSPF extended to GMPLS
  - KOM-RSVP likewise
- The VLSR translates GMPLS protocol events into generic pseudo-commands for the covered switches.
  - The pseudo commands are tailored to each specific vendor/architecture using SNMP, TL1, CLI, XML, or a similar protocol.
Virtual Label Switching Router (VLSR)

- Open source protocols running on PC act as GMPLS network element
- Control PCs participate in protocol exchanges and provisions covered switch according to protocol events (PATH setup, PATH tear down, state query, etc)
VLSR Abstraction

Ethernet network

GMPLS network

LSR

OSPF-TE / RSVP-TE

SNMP control

VLSR

VLSR

VLSR

VLSR

VLSR

VLSR
The NARB is an agent that represents the domain
The NARB performs intra-domain service capability dissemination
- Listens to OSPF-TE to acquire internal topology
- Builds an abstracted view of internal domain topology
- Advertises the abstracted TE topology to neighboring domains

The NARB also performs inter-domain path computation
- On request of RSVP, the NARB provides a loose hop ERO specifying the ingress points at each domain along the path.
  - RSVP then expands the ERO to include a strict hop path internal to the local domain.

NARB provides for resource scheduling thru the 3D-TEDB
- The Traffic Engineering DataBase (TEDB) is expanded to include a temporal dimension and an authorization (policy) dimension.
Network Aware Resource Broker (NARB) Functions – IntraDomain

- IGP Listener
- Path Computation
- Scheduling
- Edge Signaling Authentication
- Edge Signaling Enforcement
- ASTDL Induced Topology Computations
- Authorization (flexible policy based)
- Authentication
- Accounting

Diagram:
- NARB
- ASTDL Scheduling Authentication Authorization Accounting
- Edge Signaling Authorization
- IP Control Plane
- End System
- Data Plane LSP
- Ingress LSR
- Egress LSR
- Data Plane
- End System
- AS#
Network Aware Resource Broker (NARB) Functions - InterDomain

Transport Layer Capability Set Exchange
Advanced Path Computation

• **Constrained Shortest Path First**
  – Minimize blocking of light paths
  – PC for physical impairments – route for regen, route to translator to escape blocking
  – PC for asymmetric paths – i.e. xmit and recv take different paths, or different colors

• **Inter-domain abstraction of intra-domain topology**
  – Internal topology is summarized for external PC
    • Configurable from full topology to single point “black box”
- User defined summarization level maintains privacy
- Summarization impacts optimal PC but allows the domain to choose (and reserve) an internal path
Where are we?

- **GMPLS Control Plane Overview**
  - Fundamentals, MPLS Protocols, GMPLS Extensions, Standards, Interoperability, Notes
- **GMPLS Implementations**
  - Application to R&E Community
  - Research Testbeds Overview
  - DRAGON/HOPI Architecture
- **OnRamp to GMPLS R&E Networks**
  - DRAGON Software Tutorial and Demo
  - Connecting to HOPI
OnRamp to GMPLS
R&E Networks

• **DRAGON Software Tutorial and Demo**
  – Open-source GMPLS protocol stack
  – Building, configuring, using

• **GMPLS LSP provisioning demo**
  – Manchester, UK to Los Angeles, CA

• **Connecting to HOPI**
  – Data plane options
  – Control plane options
DRAGON Software

- Overview
- History
- Downloading
- Building
- Configuring
- Running
DRAGON Software: Overview

- **Open-source GMPLS Control Plane**
  - LSC, L2SC, multi-region next
  - Virtual Label Swapping Router – **VLSR**
    - Open-source OSPF-TE & RSVP-TE to control Ethernet switches and fiber switches
  - **Network Aware Resource Broker** – **NARB**
    - Performs the inter-domain routing, AAA, scheduling, PC
  - **Advanced Constrained Shortest Path First Path Computation Element** – **CSPF PCE**
    - Domain selectable abstraction levels and end-to-end LSP

- **Application Specific Topologies - **AST**
  - Formalization of application’s resource requirements – especially the network resources.
DRAGON Software:
History

- **VLSR**
  - RSVP Signaling module
    - Originated from Martin Karsten’s C++ KOM-RSVP
    - Extended to support RSVP-TE (RFC 3209)
    - Extended to support GMPLS (RFC 3473)
    - Extended to support Q-Bridge MIB (RFC 2674)
      - For manipulation of VLANs via SNMP (cross-connect)
      - Interoperates with Dell PowerConnect, Extreme, Intel, Raptor switches, etc. (switches that support RFC 2674)
  - Extended to support VLAN control through CLI
    - For switches that do not yet support RFC 2674
    - Currently interoperates with Force10 / FTOS

- **OSPF Routing module**
  - Originated from GNU Zebra
    - Zebra’s OSPF daemon originally from John Moy (in C)
  - Extended to support OSPF-TE (RFC 3630)
  - Extended to support GMPLS (RFC 4203)
DRAGON Software: History

• **NARB / CSPF PCE**
  - Written from scratch in C++ by DRAGON team
  - Lead designers and developers from USC/ISI-E

• **AST**
  - Designed to build topologies based on the needs of an application, not just individual P2P circuits

• **DRAGON management module**
  - Provides CLI / XML API to users or applications
    • Provides path setup / teardown / query functions
    • Upon path setup:
      - Computes path via NARB / CSPF PCE
      - Integrates with RSVP signaling module
    • Mostly written from scratch
      - CLI code borrowed from OSPF module’s CLI
DRAGON Software: Downloading

• Several options for downloading:
  – Stable releases
    • May not contain the latest features/bugfixes
    • …but we know it works on certain platforms
  – Bleeding-edge releases from version control
    • Does contain the latest features/bugfixes
    • May not be stable in your environment...

• Visit our website for the latest information:
  – http://dragon.maxgigapop.net
    • Click on Software on the left bar
    • Will take you to pages where you can find more information about downloading stable releases or getting repository access
DRAGON Software: Building and Installing

- **Tested platforms:**
  - FreeBSD 4.11 (gcc 2.95)
    - Should work on earlier 4.x versions
    - FreeBSD 5.x has compilation issues (gcc 3.4.2)
  - Linux 2.4.x (most testing with 2.4.20)
    - gcc 2.95 or gcc 3.2
  - Linux 2.6.x
    - gcc 3.4 - requires newer revision from the repository

- **See the INSTALL file in the release**
  - The do_build.sh script does everything except the install, so please try it first:
    - `$ ./do_build.sh`
  - The do_install script then takes care of installing everything:
    - `$ sudo ./do_install.sh`
DRAGON Software:
Building and Installing

• See the INSTALL file in the release
  – If you prefer to perform the installation manually, the following should work, assuming the net-snmp header files can be found in /usr/local/include:

```bash
$ cd kom-rsvp
$ ./configure --prefix=/usr/local/dragon --with-snmp=/usr/local
$ gmake
$ sudo gmakeinstall
$ cd ../zebra
$ ./configure --prefix=/usr/local/dragon --enable-dragon
$ make
$ sudo make install
```
DRAGON Software: Configuration

- VLSR Implementation Guide
  - DRAFT Version available on DRAGON’s Software page
  - Example topology and config files
  - Additional example configs available upon request
  - Better documentation is forthcoming…
MAX’s VLSR Lab Configuration, v6
9/2/2005
Data as per Chris Tracy
Diagram by Steve Thorpe

Example: The Label Switch Path with LSP ID=4321 and Tunnel ID=1234 could be set up from 10.100.0.20 to 10.100.0.1. GMPLS signaling would propagate from kip through various GRE tunnels over to narb1. By using simply calls during the GMPLS signal propagation, DRAGON running on the vlsr boxes would request the physical port connections on the switches that correspond to DRAGON's logical TE addresses. After the end-to-end LSP is set up, the eth1/NCIS on the end systems kip/narb1 could effectively talk directly to each other over the data path, using e.g. 192.168.100.x addressing of their own choosing.

GMPLS routing is done with an enhanced ZEBRA, GMPLS signaling done with an enhanced KOM RSVP implementation.
GMPLS LSP
Provisioning Demo

• Configure data-plane IP addresses on end hosts
  – GMPLS does not take care of this for us!

• Setup Path:
  – Kashima, Japan to Los Angeles, CA
  – Show LSP details
  – Resulting VLAN Config after Setup
  – Confirm LSP Setup using ping
  – tcpdump of control plane activity - path / resv

• Tear down demo LSP
  – Resulting VLAN Config after Delete
  – Confirm LSP Deletion using ping
GMPLS LSP Provisioning Demo Topology
GMPLS LSP Provisioning Demo

Configure data-plane IPs on end hosts

• In this case, GMPLS will provide a layer 2 connection between these hosts
  – It is up to us to configure the end systems to be able to talk to each other

[root@pc3-losa ~]# /sbin/ifconfig eth1 inet 140.173.98.28 netmask 255.255.255.224 broadcast 140.173.98.31
[root@pc3-losa ~]# /sbin/ifconfig eth1
eth1   Link encap:Ethernet   HWaddr 00:12:79:90:61:E9
       inet addr:140.173.98.28  Bcast:140.173.98.31  Mask:255.255.255.224
       inet6 addr: fe80::212:79ff:fe00:61e9/64 Scope:link
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:116 errors:0 dropped:0 overruns:0 frame:0
       TX packets:98 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:7994 (7.8 KiB)  TX bytes:7928 (7.7 KiB)
       Interrupt:217

[root@vs12000-07 ctrycy]# /sbin/ifconfig eth1
eth1   Link encap:Ethernet   HWaddr 00:58:45:68:AA:78
       inet addr:140.173.98.30  Bcast:140.173.98.31  Mask:255.255.255.224
       UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
       RX packets:191 errors:0 dropped:0 overruns:0 frame:0
       TX packets:40438 errors:0 dropped:0 overruns:0 carrier:0
       collisions:0 txqueuelen:1000
       RX bytes:19568 (19.1 KiB)  TX bytes:612455275 (564.0 Mb)
       Interrupt:27 Memory:fe8010000-fe020000

[root@vs12000-07 ctrycy]#
GMPLS LSP Provisioning Demo
Path Setup

Kashima, Japan to Los Angeles, CA connection parameters:
Source: Chicago HOPI VLSR, untagged port 38
Destination: Los Angeles HOPI VLSR, untagged port 34
Bandwidth: GigE (1,000 Mbps), Switching type: Layer-2
GMPLS LSP Provisioning Demo
List and/or Delete LSPs

Kashima, Japan to Los Angeles, CA
- Path is In-service
- Select checkbox and click Delete LSPs to tear down
Kashima, Japan to Los Angeles, CA

- Display VLAN configuration on HOPI’s Force10 switches
- For this LSP, the DRAGON NARB has assigned VLAN 3021
  - We requested ‘any’ VLAN tag on the provision form

```
LOSAR-Force10# sh vlan id 3021
Codes: * - Default VLAN, G - GVRP VLANs

  NUM  Status   Q Ports
  3021  Active  T Te 1/0
         U Gi 2/2

STTL-Force10# sh vlan id 3021
Codes: * - Default VLAN, G - GVRP VLANs

  NUM  Status   Q Ports
  3021  Active  T Te B/1
         T Te 1/0

CHIN-Force10# sh vlan id 3021
Codes: * - Default VLAN, G - GVRP VLANs

  NUM  Status   Q Ports
  3021  Active  T Te B/1
         U Gi 2/0
```
GMPLS LSP Provisioning Demo

Confirm LSP setup using ping

Kashima, Japan to Los Angeles, CA
- Ping was started before clicking ‘Provision New Circuit’ button
- Takes a few seconds to provision the LSP...

```
[root@pc3-losa ~]# ping 140.173.98.30
PING 140.173.98.30 (140.173.98.30) 56(84) bytes of data.
From 140.173.98.28 icmp_seq=1 Destination Host Unreachable
From 140.173.98.28 icmp_seq=2 Destination Host Unreachable
From 140.173.98.28 icmp_seq=3 Destination Host Unreachable
64 bytes from 140.173.98.30: icmp_seq=4 ttl=64 time=1478 ms
64 bytes from 140.173.98.30: icmp_seq=5 ttl=64 time=479 ms
64 bytes from 140.173.98.30: icmp_seq=6 ttl=64 time=238 ms
64 bytes from 140.173.98.30: icmp_seq=7 ttl=64 time=238 ms
64 bytes from 140.173.98.30: icmp_seq=8 ttl=64 time=238 ms
64 bytes from 140.173.98.30: icmp_seq=9 ttl=64 time=238 ms
64 bytes from 140.173.98.30: icmp_seq=10 ttl=64 time=238 ms
--- 140.173.98.30 ping statistics ---
11 packets transmitted, 7 received, +3 errors, 36% packet loss, time 10004ms
rtt min/avg/max/mdev = 238.900/450.378/1478.706/427.903 ms, pipe 4
```
GMPLS LSP Provisioning Demo
tcpdump of control plane activity

- PATH message from the perspective of Seattle HOPI node

```
[root@pc2-sttl tcpdump-3.9.4]# /usr/local/sbin/tcpdump -vvv -s 1500 -i gre1 ip proto 46
tcpdump: listening on gre1, link-type LINUX_SLL (Linux cooked), capture size 1500 bytes
16:41:09.607974 IP (tos 0x0, ttl 49, id 56584, offset 0, flags [none], proto: RSVP (46), length: 228) 10.100.30.33 > 10.100.30.34:
    RSVPv1 Path Message (1), Flags: [Refresh reduction capable], length: 288, ttl: 49, checksum: 0xa364
    Session Object (1) Flags: [reject if unknown], Class-Type: Tunnel IPv4 (7), length: 16
    IPv4 Tunnel Endpoint: 10.100.40.233, Tunnel ID: 0x0022, Extended Tunnel ID: 233.20.100.10
    0x0000: 0a64 28e9 0000 0022 e914 640a
    RSVP Hop Object (3) Flags: [reject if unknown], Class-Type: IPv4 plus opt. TLVs (3), length: 20
    Previous/Next Interface: 10.100.30.33, Logical Interface Handle: 0x00000003
    0x0000: 0a64 1e21 0000 0003 0000 0a64 1e25
    Time Values Object (5) Flags: [reject if unknown], Class-Type: 1 (1), length: 8
    Refresh Period: 30000ms
    0x0000: 0000 7530
    Label Request Object (19) Flags: [reject if unknown], Class-Type: Generalized Label (4), length: 8
```
GMPLS LSP Provisioning Demo

tcpdump of control plane activity

• RESV message from the perspective of Seattle HOPI node
GMPLS LSP Provisioning Demo
Tear down demo LSP

Kashima, Japan to Los Angeles, CA
- Path is In-service
- Select checkbox and click Delete LSPs to tear down

**DRAGON Network Control and Provisioning System**

<table>
<thead>
<tr>
<th>LSP Name</th>
<th>Source IP</th>
<th>Source Local ID</th>
<th>Bandwidth</th>
<th>Switching Type</th>
<th>VTAG</th>
<th>Delete</th>
</tr>
</thead>
<tbody>
<tr>
<td>kashima-to-losapc3</td>
<td>chin-hopi-vlslr (10.100.20.233)</td>
<td>Untagged Port 38</td>
<td>GigE (1000.00 Mbps)</td>
<td>Layer-2 Switch Capable (L2SC)</td>
<td>any</td>
<td></td>
</tr>
<tr>
<td></td>
<td>losa-hopi-vlslr (10.100.40.233)</td>
<td>Untagged Port 34</td>
<td>Ethernet</td>
<td>Ethernet</td>
<td>In service</td>
<td></td>
</tr>
</tbody>
</table>
GMPLS LSP Provisioning Demo
Resulting VLAN Config after Delete

Kashima, Japan to Los Angeles, CA
- Display VLAN configuration on HOPI’s Force10 switches
- For this LSP, the DRAGON NARB has assigned VLAN 3021
  - We requested ‘any’ VLAN tag on the provision form

```plaintext
STTL-Force10#sh vlan id 3021
Codes: * - Default VLAN, G - GVRP VLANs

<table>
<thead>
<tr>
<th>NUM</th>
<th>Status</th>
<th>Q Ports</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>3021</td>
<td>Inactive</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CHIN-Force10#sh vlan id 3021

Codes: * - Default VLAN, G - GVRP VLANs

<table>
<thead>
<tr>
<th>NUM</th>
<th>Status</th>
<th>Q Ports</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>3021</td>
<td>Inactive</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LOS-A-Force10#sh vlan id 3021

Codes: * - Default VLAN, G - GVRP VLANs

<table>
<thead>
<tr>
<th>NUM</th>
<th>Status</th>
<th>Q Ports</th>
<th>Ports</th>
</tr>
</thead>
<tbody>
<tr>
<td>3021</td>
<td>Inactive</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```
GMPLS LSP Provisioning Demo
Confirm LSP Deletion with ping

Kashima, Japan to Los Angeles, CA
- Ping was started before clicking ‘Delete LSPs’ button
- Takes a few seconds to delete the LSP...

```
[root@pc3-losa ~]# ping 140.173.98.30
PING 140.173.98.30 (140.173.98.30) 56(8+0) bytes of data.
64 bytes from 140.173.98.30: icmp_seq=0 ttl=64 time=238 ms
64 bytes from 140.173.98.30: icmp_seq=1 ttl=64 time=238 ms
64 bytes from 140.173.98.30: icmp_seq=2 ttl=64 time=238 ms

[...hangs here...]

--- 140.173.98.30 ping statistics ---
43 packets transmitted, 3 received, 93% packet loss, time 41997ms
rtt min/avg/max/mdev = 238.975/238.985/238.994/0.007 ms, pipe 2
```
Connecting to HOPI

- Data plane options
- Control plane options
- Introducing John Moore
  - MCNC
Data Plane Options

- Direct connection to a HOPI node
- MPLS path via Abilene
Internet 2’s Hybrid Optical Packet Infrastructure (HOPI)
Deployment evolution

Abilene nodes

10 Gbs Ethernet

OC192 Sonet/SDH (?)
Direct HOPI connection

- HOPI nodes are currently located in Los Angeles, Seattle, Chicago and Washington, DC
  - New York City coming soon
- If you can get to our PoP we can provide a 10GE or GE connection to the HOPI Force 10 switch (or via the Glimmerglass optical switch)
- Not much room for co-location, but come talk to us about your specific needs
The HOPI Node Architecture

- Force10 E600 Ethernet Switch
- Glimmerglass Fiber Switch
- Abilene
- Regional Connectors
- Other Switching Hardware (TBD)
- Performance PC
- Measurement PC
- Control PC
- Hewlett Packard

Connections:
- 10GE East
- 10GE West
MPLS path via Abilene

• The Abilene routers at each HOPI node are considered HOPI resources and can be used to originate and terminate MPLS Layer 2 tunnels to HOPI

• The tunnels need to support some flavor of Ethernet encapsulation in order to support an end-to-end Layer 2 path
  – The goal is to pass Ethernet frames transparently between the HOPI switch and your testbed

• All Abilene interior links (and a number of GigaPoP links) currently support MPLS and RSVP
  – We can turn your GigaPoP interface on too!
Requirements to build an MPLS L2 tunnel to a HOPI node

• You need a router at the remote site that supports:
  – MPLS and RSVP to originate and terminate the LSP pair (each LSP is unidirectional). In MPLS-speak these are PE (Provider Edge) routers.
  – Layer 2 encapsulation that is compatible with the Abilene Juniper routers.

• You will probably want an Ethernet switch at the remote site to act as a CE (Customer Edge) device in order to aggregate and control the Ethernet traffic going to/from HOPI. This would likely already be part of your testbed.

• Routers along the path must support MPLS and RSVP and have it turned on. These routers will only provide transit for the LSP (i.e. they will be P (Provider) routers).
Encapsulation settings on a PE (Juniper)

```
interfaces {
  ge-1/0/0 {
    description "CCC encap to FNNI400-01 Port 2/8";
    vlan-tagging;
    mtu 9180;
    encapsulation vlan-ccc;
    unit 4084 {
      description "Cheetah-HOPI interconnect";
      encapsulation vlan-ccc;
      vlan-id 4084;
      family ccc;
    }
  }
}
```
LSP configuration on a PE (Juniper)

```plaintext
protocols {
    mpls {
        label-switched-path CHEETAH->HOPI {
            from 128.109.66.252;
            to 198.32.8.202;
            no-cspf;
            primary toWASH;
        }
    ...
    ...
    ...
    }
    path toWASH {
        128.109.41.1 strict;
        128.109.66.1 strict;
        198.86.17.66 strict;
    }
    interface all;
```
Mapping LSPs to an interface on a PE (Juniper)

protocols {
    connections {
        remote-interface-switch Cheetah-HOPI-Interconnect
        interface ge-1/0/0.4084;
        transmit-lsp CHEETAH->HOPI;
        receive-lsp HOPI->CHEETAH;
    }
}
LSPs on a PE (Juniper)

centaur@jnm20-01-pbr5# run show mpls lsp detail
Ingress LSP: 1 sessions

198.32.8.202
From: 128.109.66.252, State: Up, ActiveRoute: 0, LSPname: CHEETAH->HOPI
ActivePath: toWASH (primary)
LoadBalance: Random
Encoding type: Packet, Switching type: Packet, GPID: IPv4
*Primary toWASH State: Up
  SmartOptimizeTimer: 180
  Received RRO (ProtectionFlag 1=Available 2=InUse 4=B/W 8=Node 10=SoftPreempt): 128.109.66.2 (Label=16) 198.86.17.65 198.86.17.66

Egress LSP: 1 sessions

128.109.66.252
From: 198.32.8.202, LSPstate: Up, ActiveRoute: 0
LSPname: HOPI->CHEETAH, LSPpath: Primary
Suggested label received: -, Suggested label sent: -
Recovery label received: -, Recovery label sent: -
Resv style: 1 FF, Label in: 100000, Label out: -
Tspec: rate 0bps size 0bps peak 0bps m 20 M 1500
Port number: sender 3 receiver 13823 protocol 0
PATH rcvfrom: 128.109.41.1 (ge-1/3/0.0) 83666 pkts
Adspec: received MTU 1500
PATH sentto: localclient
RESV rcvfrom: localclient
Record route: 198.86.17.66 128.109.66.1 128.109.41.1 <self>
Total 2 displayed, Up 2, Down 0

Transit LSP: 0 sessions
Total 0 displayed, Up 0, Down 0
show mpls traffic-eng tunnels

LSP Tunnel HOPI->CHEETAH is signalled, connection is up
InLabel : POS3/0, 23
OutLabel : GigabitEthernet6/0, 100000
RSVP Signalling Info:
  Src 198.32.8.202, Dst 128.109.66.252, Tun_Id 13823, Tun_Instance 3
RSVP Path Info:
  My Address: 128.109.66.2
  Explicit Route: 128.109.41.254
  Record Route: 128.109.66.1 198.86.17.66
  Tspec: ave rate=0 kbits, burst=0 bytes, peak rate=17179869 kbits
RSVP Resv Info:
  Record Route: 128.109.41.254
  Fspec: ave rate=0 kbits, burst=0 bytes, peak rate=17179869 kbits
Stuff we haven’t tried yet...

- Multi-vendor PE-PE connections
- LDP (instead of RSVP)
- Other types of encapsulation
  - Some flavors of MPLS pseudo-wires support transparent encapsulation
    i.e. no VLAN tags needed
Engineering and Operational Issues

- Fault isolation and debugging
- Monitoring dynamic networks
  - Control versus Management plane
- Alarm fault suppression
- Common Service Definition
Future Issues

- Layer 2 label swapping
- VCAT/LCAS/GFP
- Infiniband
- Framing agnostic services
  - i.e. SMPTE292 over raw lambda
- XFPs and alien wavelengths
DRAGON Deployment Status

• Optical layer:
  – 3 optical switches
  – 9 add/drop muxes (≤ 2.4G and 10G waves)

• Ethernet layer:
  – At least 1 VLSR-enabled 1GigE switch per node
  – Upgrading to 10GigE switches this month

• Software components:
  – VLSR, NARB and CSPF PCE deployed on DRAGON and HOPI testbeds
  – CHEETAH has adapted the VLSR software to support SONET/SDH (and other features) for their own testbed
Thanks!

chris@maxgigapop.net
(301) 314-6655
http://dragon.maxgigapop.net