Hybrid Multi-Layer Network Control for Emerging Cyber-Infrastructure

Final Report

**Tom Lehman, PI
Computer Network Division
University of Southern California (USC)
Information Sciences Institute (ISI)**

**Nasir Ghani, PI
Department of Electrical & Computer Engineering
University of New Mexico (UNM)**

**Eric Boyd, PI
University Corporation for Advanced
Internet Development (UCAID)
Internet2**

1. Introduction

This document provides a Final Report for the project funded by the United States Department of Energy (DOE) Office of Science project titled “Hybrid Multi-Layer Network Control For Emerging Cyber-Infrastructures” (Hybrid-MLN). The key accomplishments and results are summarized and the major products/outputs of this project are also detailed. This project was a collaboration between University of Southern California/Information Sciences Institute, University of New Mexico, and Internet2. In addition, there was a collaboration with personnel from Energy Sciences Network and Oak Ridge National Laboratory. The Hybrid-MLN project addressed many technology areas and was highly collaborative amongst the team members. This final report summarizes the accomplishments and results from the Hybrid-MLN project and team. A summary of the specific responsibilities, tasks, and focus areas of the individual organization submitting this report is also included in Section 4.

2. Accomplishments Overview

At a high level, there were four basic task areas identified for the Hybrid-MLN project. They are identified below:

- Multi-Layer, Multi-Domain, Control Plane Architecture and Implementation, includes
 - OSCARS layer2 and InterDomain Adaptation
 - Integration of LambdaStation and Terapaths with Layer2 dynamic provisioning
 - Control plane software release
 - Scheduling, AAA, security architecture
 - Network Virtualization architecture
 - Multi-Layer Network Architecture Framework Definition
- Heterogeneous DataPlane Testing
- Simulation
- Project Publications, Reports, and Presentations

A brief overview of each of these is presented in this section. A more detailed status on all the tasks and deliverables is provided Section 3.

Multi-Layer, Multi-Domain, Control Plane Architecture and Implementation

The Hybrid-MLN team worked on several key issues related to hybrid multi-layer, multi-domain control plane and data plane technologies. This project and related work was a collaboration with personnel from DOE *UltraScience Net* (USN), DOE *Energy Sciences Net* (ESnet), and Internet2 *Abilene/Dynamic Circuit Network(DCN)/Hybrid Optical Packet Infrastructure* (HOPI) networks, and the NSF *Dynamic Resource Allocation via GMPLS Optical Networks* (DRAGON) project. Many of the control plane issues revolved around the need to develop an architecture and deployable technologies to enable dynamic provision of deterministic network paths across these networks.

Much progress was made on the development of control plane for hybrid networks on this project. An important part of this work was the development of the architecture, inter-domain protocols, and AAA (Authentication, Authorization, and Accounting) architecture to enable multi-domain, multi-layer, multi-service hybrid network environment provisioning. This work included both general design activities and specific implementation on the collaborating organization's networks. As this worked progressed, the primary focus was on dynamic provisioning across the ESnet and

Internet2 infrastructures as well as connected organizations and sites. On the design side this project's activities resulted in a completed architecture and design for inter-domain control plane exchanges between networks. A challenging issue when interoperating these networks revolved around the heterogeneous environment for both the dataplane and control planes. We chose an approach of developing web service based inter-domain operations to allow us to provide the level of abstraction necessary to operate in this environment. We have completed the design, formal specification, and implementation of this web service interdomain control plane. In order to gain broader acceptance of this architecture and work, we worked inside the DICE (DANTE, Internet2, Canarie, ESnet) Control Plane working group to define and agree on these interdomain specifications. This approach proved to be very successful and has resulted in a reasonably broad acceptance of this architecture. This capability has been realized via extension of the OSCARS system to utilize the InterDomain Controller Protocol (IDCP) developed as part of the DICE activities. The results and current status of the DICE IDCP protocol specification work can be found at www.controlplane.net. In addition, site traffic management systems LambdaStation and TeraPaths have been modified and integrated into the multi-domain environment enabled by the IDCP protocol. The DICE IDCP also included an initial AAA and Scheduling architecture which was based on an extension of the OSCARS system. This has been implemented and is in active use for multi-domain provisioning. During this project, additional work was started to expand the AAA and scheduling capabilities to allow for additional capabilities for user level authentication and authorization as well as increased flexibility in available time slot identification. It is expected that this initial work in AAA and scheduling will be used as the basis for future work in these areas as part of other projects.

Another key accomplishment in this area was the development of a Multi-Layer, Multi-Domain, Control Plane Architecture definition. This work included the development of a Multi-Layer Networking Architecture Framework document. The purpose of this document was to define a high level architecture for Multi-Layer networks in order to provide a framework, context, and organization for more detailed research efforts in the various areas. The approach taken for this architecture definition included the definition of "capability planes" for a "generic technology layer". Subsequently we looked at the individual technologies and addressed the characteristics and features that are unique to the specific technology layers. With these foundations in place we then examined the construction and operation of a multi-layer architectures that combine two or more of the technology layers. The "Multi-Layer Networking, An Architecture Framework" developed is provided in Appendix A of this report.

We also developed an architecture document which specifically addressed the implementation approach taken for the control plane which allowed interoperation between ESnet, Internet2, and other peering networks. This architecture document, "Hybrid Multi-Layer Network (Hybrid-MLN) Control Plane Architecture" is provided in Appendix B of this report. This work defined an initial architecture for multi-layer, multi-domain provisioning and subsequent implementation demonstrated the validity of the approach. This architecture included networks with the following technology layers: Packet Switching Capable (PSC) layer, Layer-2 Switching Capable (L2SC) layer, Time Division Multiplexing (TDM) layer and Lambda Switching Capable (LSC) layer. Additional details on the interdomain protocol design, status, and use is presented as part of the individual task status reported in Section 3.

Heterogeneous DataPlane Testing

A dataplane evaluation study and performance testing was completed for multi-layer eScience networks. This work was one of the first studies of its kind and comprised detailed real-world empirical measurements along with in-depth network simulation analyses. Specifically, the networks analyzed in this effort include the DOE USN, DOE ESnet, Internet2 Abilene/HOPI, and

NSF DRAGON. A comprehensive test report was released and results were also presented at multiple ESCC/Internet2 Joint Techs Meetings. The "Hybrid Multilayer Network Data Planes Network-Layer Testing & Analysis Report" and associated test plans are provided in Appendix C of this report.

Simulation

In addition, Dr. Nasir Ghani and his team from University of New Mexico led the efforts on simulation and analysis of the network architectures and associated deployed infrastructures. The key areas of investigation from a simulation and modeling perspective were architecture design, multi-layer routing algorithms, and multi-layer path selection/scheduling. This simulation and analysis work has factored in the actual topologies for ESnet SDN and Internet2 DCN as a template for this work. The project approach was to utilize the outputs from the modeling activities to evaluate current architectures and inform decisions on future designs and prospects for scaling. This simulation work is continuing via a contract extension for the University of New Mexico effort. Additional details regarding these efforts will be provided as part of Dr. Nasir Ghani's report specifically on the simulation activities. We anticipate subsequent simulation results to assist us in evaluation of the defined architecture as well as provide alternatives and improvements to consider.

Project Publications, Reports, and Presentations

There were many project related publications, reports, and presentations. These are specifically identified in Section 6. In addition, copies of project presentations are provided in Appendix D of this report.

3. Accomplishment and Deliverable Details

All Hybrid-MLN objectives, tasks, and deliverables were completed. A summary of these is provided below:

- **Inter-Domain Protocol Definition**
We have completed the design, formal specification, and implementation of this web service interdomain control plane. In order to gain broader acceptance of this architecture and work, we have worked inside the DICE (DANTE, Internet2, Canarie, ESnet) Control Plane working group to define and agree to these interdomain specifications. This has worked well and has resulted in a reasonably broad acceptance of this architecture. We refer to these results of this work as the InterDomain Controller Protocol (IDCP). The IDCP defines three independent, but related, phases of InterDomain Web Service based messaging: Topology Exchange, Resource Scheduling, Signaling. In addition, there is a User Request phase defined. The following organizations have implemented and/or deployed systems which are compatible with this IDCP:
 - Internet2 Dynamic Circuit Network (DCN) ION Service
 - ESNet Science Data Network (SDN)
 - GEANT AutoBahn System
 - Nortel (via a wrapper on top of their commercial DRAC System)
 - Surfnets (Netherlands Nortel based Solution)
 - United States Large Hadron Collider Network (USLHCNet)
 - CalTech

- NYSERNET (New York Regional Network)
- Lonestar Education And Research Network (LEARN) Texas Regional Network)
- Louisiana Optical Network Initiative (LONI) (Louisiana Regional Network)
- Northrop Grumman Corporation
- University of Amsterdam
- Mid-Atlantic Crossroads (MAX) (Washington D.C. regional network)
- Japanese Research Network (JGN2)

In addition, the following "higher level service applications" have adapted their existing systems to communicate via the user request side of the IDCP:

- LambdaStation (www.lambdastation.org)
- TeraPaths (www.racf.bnl.gov/terapaths)
- Phoebus (damsl.cis.udel.edu/projects/phoebus)

This system is now deployed on ESnet Science Data Network (SDN), Internet2 Dynamic Circuit Network (DCN), and multiple other networks and is in regular use. Additional details on the IDCP are located at www.controlplane.net.

- **OSCARS System Adaptation to other Control Planes**
The OSCARS system, originally developed by ESnet was utilized as the basis for the Web Service based InterDomain control plane messaging. This system was extended to enable multi-layer, multi-domain dynamic provisioning via implementation of the InterDomain Controller Protocol as described in the above status description. The OSCARS implementation work was accomplished jointly by ESnet and Internet2. This system is in use by ESnet, Internet2 and others as the basis for the Web Service layer for their dynamic control planes.
- **Addition of GMPLS functions to USN Control Plane**
Due to programmatic directions, we were not able to deploy the GMPLS control plane on the USN network. However, we did deploy the Web Service/GMPLS control plane on the Internet2 DCN and USLHC Network which uses the same Ciena Core Director technology as USN. So it would be possible to deploy a similar capability on the USN Network should it be desired.
- **AAA Feature Incorporation**
A design for an AAA model for dynamic provisioning in Multi-Layer, Multi-Domain Hybrid Networks was completed. This system is an extension of the original OSCARS model, and provides a mechanism for user to network and network to network authentication and policy application. This system was implemented by the OSCARS development team is in use and deployed as part of the overall control plane deployments as noted above. This system deploys a user-level authentication process based upon X.509 certificates. The associated authentication control process provides a mechanism for a simple check of user versus a preset bandwidth allocation value. In addition, we also conducted some preliminary analysis on future more robust and flexible sets of AAA mechanisms to accommodate a more adhoc mesh-based interconnection of hybrid networks. Some of the models identified for future consideration are extensions of technologies used on other projects such as PerfSonar (<http://www.perfsonar.net>) and EduGain (<http://www.edugain.org/>). The work completed as part of this project on AAA models provides a good baseline and framework for future research and development in this area.

- **Scheduling Features**

A design for a scheduling system for dynamic provisioning in Multi-Layer, Multi-Domain Hybrid Networks was completed. The scheduling system features were incorporated into the DICE IDCP and was based on an extension of the original OSCARS model. This system provides mechanisms for multi-domain scheduling of lower layer services, using a multi-stage, multi-domain scheduling process. This system was implemented by the OSCARS development team is in use and deployed as part of the overall control plane deployments as noted above. The OSCARS system uses a database to maintain a network view and process requests in the context of current and future available bandwidth by evaluating database entries. In addition, we also conducted some preliminary analysis on future more robust and flexible sets of scheduling mechanisms. A conclusion was that a richer set of scheduling and control parameters will be desired as we move forward toward a more adhoc mesh-based interconnection of hybrid networks. Additionally there needs to be more advanced mechanisms for dynamic exchange of scheduling information between domains. For multi-level networks which allow for provisioning and resource control at multiple network layers, there needs to be a mechanism to utilize the higher-level scheduling information as a constraint variable during path computation and signaling. This is quite a complex space as the number of constraints can rapidly become greater than can be efficiently processed in a reasonable amount of time. The work completed as part of this project on scheduling provides a good baseline and framework for future research and development in this area.
- **Open Source Control Plane Software Release**

Multiple versions of the open source control plane software which is currently running on ESnet SDN, Internet2 DCN, and other networks was released during the duration of this project. While the overall development of this control plane software is larger than the Hybrid-MLN project, the features addressed by the Hybrid-MLN project, such as the DICE IDCP, were included in these software releases. A link to the open source software releases can be found on the project web site, hybrid.east.isi.edu ==> Software link.
- **Data Plane - Multi-layer Circuit Performance and Characterization**

We completed detailed dataplane level testing which spanned multiple networks including DOE USN, DOE ESnet, Internet2 Abilene/HOPI, and NSF DRAGON. This work included the following tasks:

 - Developed a detailed test plan documenting the specific network configurations, test equipment placements, and data collection plans. This included the design of inter-network routes over various eScience networks (USN, ESnet, Internet2 Abilene, Internet2 HOPI, and DRAGON).
 - Coordinated loan of test equipment from SPIRENT, installed equipment in multiple Internet2 and USN network points of presences.
 - Conducted extensive testing and data collection.
 - Conducted post data collection analysis, made available via a web page, worked with project collaborators to analyze data and develop conclusions.
 - Worked closely with Dr. Nasir Ghani to correlate simulation activities based on actual test scenarios and configurations.
 - Developed a detailed test report which included summary of data collected and correlation to simulation results. This report presents full details on many of the extensive empirical tests conducted by the project team and extracts key findings and conclusions.

Overall, the findings of the data-plane performance evaluation phase confirmed the viability of building dedicated bandwidth services over mixed IP/MPLS, Ethernet, and SONET/SDH networks. The common conclusion was overall performance was well in-line with expectations if IP/MPLS tunnels were not driven over 98% of their capacity. However, SONET/SDH links performed exceptionally well and maintained solid delay jitter/loss performance throughout the loading regime. The "Hybrid Multilayer Network Data Planes Network-Layer Testing & Analysis Report" and associated test plans are provided in Appendix C of this report.

- **Multi-Layer, Multi-Domain Network Provisioning**
We completed an initial architecture definition to accomplish routing, path computation, and signaling required to provision across the various types of layer transitions. This includes transitions between PSC (router), L2SC (ethernet), TDM (SONET/SDH), and LSC (lambda) regions. In this initial control plane development we have normalized all interdomain links at the Ethernet VLAN level and required any layer transitions to occur inside domains in a manner invisible to external entities. Technology layer hiding as part of the domain abstraction process proved to be the most prudent method to allow initial progress in this area. The results of this work are reflected in the designs and implementations associated with the IDCP and dynamic circuit network deployments on the research and education networks. The work completed as part of this project in this area provides a good baseline and framework for future research and development activities. Additional details on the multi-layer, multi-domain provisioning architecture developed is provided in the "Hybrid Multi-Layer Network (Hybrid-MLN) Control Plane Architecture" document located in Appendix B of this report.
- **Integration of QoS protected Best Effort Flows into End-to-End Hybrid Paths**
This objective was completed via working with the Terapaths (www.racf.bnl.gov/terapaths) and LambdaStation (www.lambdastation.org) groups. We worked with these groups to develop architectures and designs for the integration of end-site QoS flows into the wide area multi-domain lower layer service provisioning enabled via our previously described project work. They subsequently modified their applications to interface with the Web Service based Hybrid Network architecture described earlier. This has enabled the Terapaths and LambdaStation management of IP routed flows to be integrated with the lower layer hybrid network services. This is currently in active use at several ESnet SDN and Internet2 DCN connected sites.
- **Network Virtualization and Architecture and Experimentation**
This project's main accomplishment in the area of network virtualization was the development of the key building block in the form of the multi-domain, lower layer circuit provisioning. This capability is the "atomic function" which forms the basis to evolve from basic single circuit provisioning actions to a broader, multiple constraint topology and network virtualization provisioning. We identified that the objectives for these scenarios would be to coordinate multiple atomic actions in order to invoke a true "virtual" network environment. The work completed in this area was a preliminary analysis of the feasibility of using the base functions provided by the IDCP discussed earlier. It is concluded that these functions can provide the basis for future network virtualization function development. Due to project task redirection associated with the Multi-Layer Network Architecture Framework Definition, additional detail designs for network virtualization were not addressed.

- **Control Plane Security Architecture and Implementation**
The initial security architecture and implementation for the hybrid multi-domain network control plane was based on web service security, secure socket layer (SSL), and X.509 certificates. This architecture was an extension of the OSCARS framework, and extended to apply to the multi-domain environment and the IDC Protocol discussed earlier. This architecture proved to be sufficient to allow initial progress in the area of security for multi-domain hybrid network provisioning. It is concluded that these functions can provide the basis for future hybrid network security, but additional study is needed to address more robust mechanisms for security monitoring, logging, and analysis. The work completed as part of this project on security provides a good baseline and framework for future research and development in this area. Due to project task redirection associated with the Multi-Layer Network Architecture Framework Definition, additional detail designs for control plane security were not addressed.
- **Multi-Layer Network Architecture Framework Definition**
The purpose of this task was to develop an architecture and framework for Multi-Layer network from a functional and service perspective. Another purpose of this document was to define a high level architecture for Multi-Layer networks in order to provide a framework, context, and organization for more detailed research efforts in the various areas. The approach taken for this architecture definition included the definition of “capability planes” for a “generic technology layer”. Subsequently we looked at the individual technologies and addressed the characteristics and features that are unique to the specific technology layers. With these foundations in place we can then examined the construction and operation of a multi-layer architectures that combine two or more of the technology layers. The "Multi-Layer Networking, An Architecture Framework" developed is provided in Appendix A of this report.

4. Organization Activities

This project was a collaboration between University of Southern California/Information Sciences Institute, University of New Mexico, and Internet2. In addition, there was a collaboration with personnel from Energy Sciences Network and Oak Ridge National Laboratory. The Hybrid-MLN project addressed many technology areas and was highly collaborative amongst the team members. As a result of this close collaboration, all of the tasks included contributions from multiple team members. However, the general task and work areas can be identified as shown below.

University of Southern California/Information Sciences Institute activities were focused on the areas of multi-layer/multi-domain control plane architecture definition, web service based multi-domain protocol definition, integration of IP routed paths into the layer2 environments, and data plane testing. University of Southern California/Information Sciences Institute also coordinated the generation of the multi-layer architecture documents.

University of New Mexico activities focused on data plane testing analysis/simulation/reporting, simulation of control plane architecture and performance, and multi-layer/multi-domain control plane architecture definition.

Internet2 activities were focused on the areas of web service based multi-domain protocol definition, interdomain protocol implementation, integration of IP routed paths into the layer2 environments, and data plane testing. The data plane testing activities included support to locate

test equipment in Internet2 points of presence to allow connection to operational networks for data collection.

5. Project Presentations and Publications

The Hybrid-MLN project generated multiple presentations, publications, and other documents. These are listed below and copies are provided in appendices to this report as indicated.

5.1. Project Presentations

The project collaboration produced multiple presentations. These are listed below and copies are provided in Appendix D.

- Multi-Layer Networking An Architecture Framework
Presentation at ESnet Site Coordinating Committee (ESCC)
July 22-23, 2009
Indianapolis, Indiana
Tom Lehman, (USC/ISI)
Chin Guok (ESnet)
Brian Tierney (ESnet)
Nasir Ghani (UNM)
- Network Convergence: Looking Forward
Presentation at Networking Research Challenges Workshop
September 28-30, 2008
Seattle, Washington
Tom Lehman, (USC/ISI)
- Hybrid Network Architectures: A Forward-Looking Vision
Presentation at January 08 ESCC/Joint Techs
Tom Lehman, (USC/ISI)
Xi Yang, (USC/ISI)
Chin Guok, Energy Sciences Network (ESnet)
John Vollbrecht, Internet2
Andrew Lake, Internet2
Nasir Ghani, Tennessee Tech University
Nagesware Rao, Oak Ridge National Laboratory
- Heterogeneous Data Plane Testing on USN, ESnet, and Internet2 Networks
Presentation at July 07 ESCC/Joint Techs
Nasir Ghani, Tennessee Tech University
Chin Guok, Energy Sciences Network (ESnet)
Tom Lehman, University of Southern California
John Moore, ITEC - North Carolina
Nagesware Rao, Oak Ridge National Laboratory
John Vollbrecht, Internet2
- Hybrid Network Control Plane Interoperation Between Internet2 and ESnet
Presentation at July 07 ESCC/Joint Techs
Chin Guok, Energy Sciences Network (ESnet)
Andrew Lake, Internet2

Tom Lehman, University of Southern California
John Vollbrecht, Internet2

- Multi-Layer, Multi-Domain Hybrid Network Interoperation and Performance Presentation at Feb 07 ESCC/Joint Techs
Tom Lehman (USC/ISI)
Nasir Ghani (Tenn Tech)
Chin Guok (ESnet)
Nagi Rao (ORNL)
John Vollbrecht (Internet2)
John Moore (MCNC)

5.2. Project Documents

The project collaboration produced the follow project documents. These are listed below and copies are provided in Appendices as indicated.

- DataPlane Testing and Evaluation (Appendix C)
 - Hybrid Network DataPlane Test Plan
 - Hybrid Network Test Configuration
 - Hybrid Network Dataplane Test Report
- Multi-Layer Networking, An Architecture Framework (Appendix A)
- Hybrid Multi-Layer Network (Hybrid-MLN) Control Plane Architecture (Appendix B)

5.3. External Publications

Several technical publications have been generated based upon some of the work in this project, pertaining to both multi-layer data and control plane. These are listed below.

- Qing Liu (Oak Ridge National Laboratory), Tannous Frangieh(Virginia Tech), Feng Xu (University of New Mexico), Chongyang Xie (University of New Mexico), Nasir Ghani (University of New Mexico), Thomas Lehman (USC/ISI), Chin Guok (Energy Sciences Network), IEEE Globecom 2009 Optical Networks and Systems Symposium (GC'09 ONS), 30 November-4 December 2009, Honolulu, Hawaii, USA
- Inter-Domain Routing Scalability in Optical DWDM Networks, Liu, Q.; Xie, C.; Frangieh, T.; Ghani, N.; Gumaste, A.; Rao, N.; Lehman, T.; Computer Communications and Networks, 2008. ICCCN '08. Proceedings of 17th International Conference on 3-7 Aug. 2008 Page(s):1 - 7, Digital Object Identifier 10.1109/ICCCN.2008.ECP.22
- Control plane design in multidomain/multilayer optical networks, Ghani, N.; Qing Liu; Benhaddou, D.; Rao, N.S.V.; Lehman, T.; Communications Magazine, IEEE Volume 46, Issue 6, June 2008 Page(s):78 - 87, Digital Object Identifier 10.1109/MCOM.2008.4539470
- Multi-Domain Multi-Granularity Service Provisioning in Hybrid DWDM/SONET Networks, Liu, Q.; Ghani, N.; Rao, N.S.V.; Lehman, T.; High-Speed Networks Workshop, 2007 11-11 May 2007 Page(s):26 - 30, Digital Object Identifier 10.1109/HSNW.2007.4290540

- Measurements On Hybrid Dedicated Bandwidth Connections, Rao, N.S.V.; Wing, W.R.; Qishi Wu; Ghani, N.; Qing Liu; Lehman, T.; Guok, C.P.; Dart, E.; High-Speed Networks Workshop, 2007 11-11 May 2007 Page(s):41 - 45, Digital Object Identifier 10.1109/HSNW.2007.4290543
- Control Plane Architecture and Design Considerations for Multi-Service, Multi-Layer, Multi-Domain Hybrid Networks, Lehman, T.; Xi Yang; Guok, C.P.; Rao, N.S.V.; Lake, A.; Vollbrecht, J.; Ghani, N.; High-Speed Networks Workshop, 2007, 11-11 May 2007 Page(s):67 - 71, Digital Object Identifier 10.1109/HSNW.2007.4290549
- Q. Liu, C. Xie, T. Frangieh, N. Ghani, A. Gumaste, N. S. V. Rao, "Routing Scalability in Multi-Domain DWDM Networks", Photonic Network Communications, Vol. 17, No. 1, February 2009, pp. 63-74.
- Q. Liu, N. Ghani, N. Rao, A. Gumaste, M. Garcia, "Distributed Inter-Domain Lightpath Provisioning in the Presence of Wavelength Conversion," Computer Communications, Vol. 30, No. 18, December 2007, pp. 3362-3375.

5.4. Websites & Repositories

A main project website was created, detailing the project's goals, PI participants, and accomplishments (e.g., technical reports, repositories, etc). This is available via the following website URL: <http://hybrid.east.isi.edu>.

Appendix A

Multi-Layer Networking, An Architecture Framework

Multi-Layer Networking

An Architecture Framework

Table of Contents

1	Introduction	3
2	Service Workflows.....	6
3	ApplicationPlane.....	9
4	ServicePlane.....	11
5	AAPlane.....	14
6	ControlPlane	16
7	ManagementPlane.....	18
8	DataPlane	20
9	Specific DataPlane Types	22
9.1	PSC.....	23
9.2	L2SC	24
9.3	TDM.....	24
9.4	LSC	24
9.5	FSC.....	25
10	Multi-Layer Architectures.....	25
10.1	Multi-Layer Networking - Vertical	25
10.2	Multi-Layer Networking - Horizontal	26
10.3	Multi-Layer Networking - Combined.....	27
10.4	Multi-Layer Networking - InterDomain	28
10.5	Hybrid Networking (Multi-Layer Traffic Engineering and Traffic Grooming)	28
10.6	Nested Capability Planes	28
11	Glossary	29
12	Authors.....	30
13	Acknowledgements	30
	Appendix 1 Standards Bodies Overview	31
	Appendix 2 References.....	33

Multi-Layer Network Architecture

1 Introduction

This document presents an architecture description for a “Multi-Layer Network”. In this context the term “Multi-Layer” refers to fact that modern networks may use multiple technologies to implement the desired network services. For example, operators can use technologies such as Internet Protocol (IP), multi-protocol label switching (MPLS), Transport-MPLS (T-MPLS), Ethernet, Ethernet Provider Backbone Bridge Traffic Engineering (PBB-TE), Synchronous optical networking (SONET)/Synchronous Digital Hierarchy (SDH), next-generation SONET/SDH, and/or Wavelength-Division Multiplexing (WDM).

The purpose of this document is to define a high level architecture for Multi-Layer networks. The objective of this architecture is to provide a framework, context, and organization for more detailed research efforts in the various areas identified. The more detailed research efforts are expected to result from current and future research projects.

In this document we take the approach of first defining the “capability planes” for a “generic technology layer”. Subsequently we look at the individual technologies and address the characteristics and features that are unique to the specific technology layers. With these foundations in place we can then examine the construction and operation of a multi-layer architectures that combine two or more of the technology layers.

“Capability planes”, or CapabilityPlanes, represent a grouping or layering of associated functions that are required for construction and operation of a single technology layer. The following CapabilityPlanes are identified:

- DataPlane

The DataPlane is the set of network elements which receives, sends, and switches the network data. For this architecture definition we identify the dataplane options in terms of “technology regions”. A “technology region” is a set of network elements which are grouped together and utilize the same dataplane “technology type”. The technology types are defined using the standard Generalized Multi-Protocol Label Switching (GMPLS) [1] nomenclature of Packet Switching Capable (PSC) layer, Layer-2 Switching Capable (L2SC) layer, Time Division Multiplexing (TDM) layer, Lambda Switching Capable (LSC) layer, and Fiber-Switch Capable (FSC). Additionally, we associate these technology types with the more common terminology of Layer 0, Layer 1, Layer 2, Layer 3, etc. Details on the DataPlane technology types, descriptions and features are provided in the Section 8 (DataPlane).

- ControlPlane

The ControlPlane plane is responsible for routing, path computation, and signaling functions associated with control of the DataPlane. This generally includes maintaining topology information and configuring network elements in terms of data ingress, egress, and switching operations. The ControlPlane is one of two CapabilityPlanes which directly interacts with the DataPlane.

- ManagementPlane

The ManagementPlane refers to the set of systems and processes that are utilized to monitor, manage, and troubleshoot the network. This includes functions to support user services as well as network maintenance, upgrades, and reconfigurations. This plane is responsible for collecting data and monitoring of the network. In addition, this CapabilityPlane may also include capabilities to configure the network elements with the support of the control plane or via independent actions. The ManagementPlane is one of two CapabilityPlanes which directly interacts with the DataPlane.

- AAPlane

This plane is the Authentication and Authorization plane. This plane is responsible for the mechanisms which allow the other planes to identify and authenticate users and receive associated policy information.

- ServicePlane

The ServicePlane refers to the set of systems and processes that are geared towards providing services to users and maintaining state on those services. The Service plane will generally rely on the functions of the

ControlPlane and/or ManagementPlane to effect actual changes on the DataPlane. In addition, the ServicePlane will typically maintain databases on current and future service instantiations and coordinate associated workflow processes.

- ApplicationPlane

The ApplicationPlane provides higher level functions which will generally be tailored for domain specific purposes. It is envisioned that the ApplicationPlane is the area where domain specific experts will be creative and develop capabilities which are specific and unique to their application requirements. The ApplicationPlane will rely on the capabilities offered by one or more ServicePlanes to accomplish its objectives. In this context, the boundary between the ApplicationPlane and ServicePlane is the network demarcation where a detailed and specific network service interfaces will be defined.

Figure 1 depicts a concept of operation for how these capability planes interact. As can be seen from this figure, the only planes which actually touch the DataPlane are the ControlPlane and ManagementPlane. In general, the ServicePlane relies on the capabilities of the ControlPlane and ManagementPlane to effect changes in the DataPlane. The AAPlane is a support plane for which all the other planes may interact with directly to accomplish their specific functions. The ApplicationPlane provides higher level functions which will generally be tailored for domain specific needs. In this context the boundary between the ApplicationPlane and ServicePlane defines the network demarcation where the network service interfaces would be accessed. Figure 2 depicts another view of the relationships between the planes which is more similar to a traditional protocol layer view.

Figure 1 Capability Planes – Graphical View

Figure 2 Capability Planes - Layered View

A very important concept in the definition of this architecture is that of the single network “technology region” as described in the DataPlane overview presented above. All other CapabilityPlanes are defined in the context of their responsibilities with respect to a single network technology region. An actual network instantiation may consist of one or more technology regions interconnected in a horizontal and/or vertical manner. Additional details regarding multi-layer network construction are presented in Section 9 (Multi-Layer Architectures). The key point to note for the following specific CapabilityPlane sections is that they are described in the context of a single technology region. An example of a technology region would be a group of interconnected routers which would be a PSC technology region, or a group of WDM switching elements which would be a LSC technology region.

Another important item to note is that while this architecture definition maintains a distinct single CapabilityPlane to single DataPlane relationship, it is recognized that in real world implementations, this may not be the case. For instance, if one constructs a network with Routers (PSC) on top of WDM (LSC) equipment, there may be a desire to have a single CapabilityPlane which is responsible for both the PSC and LSC technology region. This would be acceptable within the framework of this architecture description. The purpose of the multi-layer architecture described herein is to define functions and capabilities. Implementation options such as combining CapabilityPlanes into a single instantiation is compatible with the concepts presented here.

Another way to view the DataPlane layer to CapabilityPlane relationship is from a more DataPlane layer centric view. Figure 3 depicts this type of view where the DataPlane layer options are identified in terms of Layer 1, Layer 2, and Layer 3.

Figure 3 Layer View of Capabilities

In this figure the capabilities identified for each layer are the same as the CapabilityPlanes discussed earlier. The remainder of the document will describe the architecture from a CapabilityPlane perspective. A primary reason for this is that many of the CapabilityPlane functions are independent of the specific DataPlane layer implementation. Therefore, for each CapabilityPlane we will first describe the layer independent capabilities, and then discuss the layer dependent features where they exist.

It should also be noted that the AA CapabilityPlane discussed above is distinct from security. Where the AAPlane provides authentication and authorization features, security is focused on preventing unauthorized access to, and/or damage to the network capabilities. As a result, each CapabilityPlane will have its own security features defined. This is referred to as “CapabilityPlane Security” and will be defined uniquely for each of the individual CapabilityPlanes. Since there are many methods and options regarding what/how security features are incorporated, the definition of the CapabilityPlane Security in this architecture will focus on capability and requirements, and not specific implementation methods.

In the following sections we will identify the key characteristics for each of the CapabilityPlanes. These characteristics are categorized as follows:

- Functions
- Function Interfaces
- Layer Unique Considerations
- Security Considerations

A control and/or configuration action on a network will typically be the result of a workflow process which coordinates actions across multiple CapabilityPlanes. This is referred to as a multiple CapabilityPlane workflow in this document. The modular nature of DataPlane Layers and the associated CapabilityPlanes allow for many different workflows that can be tailored to the needs and requirements of individual users and network operators. In the next section we present some example workflows as an introduction to the use and function of CapabilityPlanes as it relates to individual Layer (or DataPlane) control.

The remainder of this document is organized as follows: Section 2 presents example workflows to demonstrate the concept of coordinated CapabilityPlane processes. Section 3-8 describe each of the individual CapabilityPlanes. Section 9 provides additional details on specific DataPlane types. Section 10 provides discussion on several multi-layer network architecture concepts and design alternatives. Section 11 provides a glossary of terms utilized in this document. Appendix 1 provides a discussion of standards bodies and relationship to the concepts presented in this document. Appendix 2 provides a reference list.

2 Service Workflows

Coordinated action across several CapabilityPlanes requires workflow processes. There are potentially multiple types of workflows, each involving all or a subset of the CapabilityPlanes. Several example workflows are presented below. These are intended to show high level views of possible workflows, and are not intended to show every detail or every possible workflow.

Figure 4 depicts a very simple workflow associated with a user requesting a circuit instantiation across a single network domain or region. In this scenario, the user request initiates a ServicePlane coordinated workflow which requires actions by the AAPlane and ControlPlane. The result is a circuit instantiation across the DataPlane.

Figure 4 Single Domain Circuit Request

Figure 5 depicts a similar service request from the user perspective as described in Figure 4. However, in this case the endpoint requested requires multi-domain provisioning actions. As a result the ServicePlane coordinated workflow includes ServicePlane inter-domain interactions which result in specific AAPlane and ControlPlane actions in their respective domains. The result is a circuit instantiation across each DataPlane which is “stitched” together to operate as a single end to end circuit from the user perspective.

Figure 5 Multi-Domain Circuit Request - ServicePlane InterDomain

Figure 6 shows a workflow which is also a multi-domain circuit instantiation service request, similar to Figure 5. However in this case the inter-domain interactions are directly between the domain ControlPlane functions. This scenario may represent a more traditional MPLS/GMPLS peering model between two network domains. From a user perspective the result is a circuit instantiation across each DataPlane identical to the service resulting from the earlier workflow.

Figure 6 Multi-Domain Circuit Request - ControlPlane InterDomain

Figure 7 depicts a workflow resulting from a request for monitoring information. In this scenario, the user may be requesting status of a specific network element, circuit, or other instantiated service. In response to the user request, the ServicePlane coordinates the actions of the AAPlane and ManagementPlane to query the DataPlane and provide the information to the user. This workflow also shows how this type of request may require multi-domain operations which are handled through ServicePlane to ServicePlane interactions.

Figure 7 Multi-Domain Monitoring and Troubleshooting Procedures

Figure 8 shows a workflow which is based on a “vertical” multi-layer interaction vs. a “horizontal” multi-layer interaction as depicted in the previous multi-region scenarios. The CapabilityPlane interactions are similar for both cases. The main distinction between vertical vs. horizontal is in the use case and application of the resulting service. For instance, in the case of the multi-domain circuit instantiation workflow from Figure 5, the resulting service is presented as an end-to-end dedicated resource network path for the user application. In the case of the vertical multi-region service provisioning shown in Figure 8, the resulting service instantiated at the lower level (Layer 1) is handed off to the higher level (Layer 3), which in turn provides the capability to provision a service at the higher layer for the user. An example of this would be if the initial bandwidth availability between two Layer 3 provider edge (PE) routers was insufficient to meet the user’s service request, resulting in the instantiation of a Layer 1 cut-through path that provided the required bandwidth for the Layer-3 user service connection.

Figure 8 Multi-Layer Vertical Service Provisioning

3 ApplicationPlane

The ApplicationPlane provides higher level functions which will generally be tailored for domain specific purposes. It is envisioned that the ApplicationPlane is the area where domain specific experts will be creative and develop capabilities which are specific and unique to their application requirements. The ApplicationPlane will rely on the capabilities offered by one or more ServicePlanes to accomplish its objectives. In this context, the boundary between the ApplicationPlane and ServicePlane is the network demarcation where a detailed and specific network service interfaces will be defined.

The set of ApplicationPlane functions defined in this section is not intended to be an exhaustive list of what will likely be found in specific ApplicationPlane instantiations. A key notion of the ApplicationPlane is that it will be tailored by domain specific experts for their needs and requirements. The functions identified in this section are intended to provide some examples of the types of ApplicationPlane functions and services which are anticipated.

Some of the services the ApplicationPlane may provide include co-scheduling of multiple “application resources” in combination with the network services obtained via the ServicePlane interface. In this context “application resources” are envisioned to be items like compute clusters, storage repositories, scientific instruments, or any other equipment which may be integral to a workflow pipeline that is associated with the network services being provided. An ApplicationPlane may need to query multiple ServicePlanes to determine service availability and then coordinate network scheduling with other resource scheduling.

Functions

The following functions are identified for the ApplicationPlane:

- **Co-Scheduler**
Responsible for contacting one or more ServicePlanes to determine if a given network service is available at the time needed. This will likely be accomplished in the context of separate actions to coordinate availability of application specific resources such as compute clusters, storage repositories, and scientific instruments.
- **Session Control**
A session is the end-to-end composition of one or more ServicePlane service instantiations. The ApplicationPlane will generate a ServicePlane request which is tailored to the application requirements. A session may be created based on application requirements such as throughput, jitter, and time period requirements. In addition, the ApplicationPlane may be concerned with higher level session related configurations which are beyond the scope of the ServicePlane and the feature set described in this architecture document. These types of features are generally application or domain specific and may involve configuration of end system applications, end system interfaces, selection of specific protocols, or other unique application configurations.
- **ApplicationPlane Status**
This function involves the maintenance of ApplicationPlane state such that recovery mechanisms can restore operation after ApplicationPlane failures. The recovery functions may be completely internal to the ApplicationPlane, or in some instances may include a coordinated effort across multiple CapabilityPlanes. In this instance, the ApplicationPlane may be providing state information as part of a larger recovery workflow process.

Function Interfaces

In general the Functional Interface for the ApplicationPlane is very similar to the ServicePlane, but the ApplicationPlane coordinates across multiple ServicePlanes. The following ApplicationPlane Functional Interfaces are defined

- **Application Request (app-req)**
This interface is the messaging exchanged between the ServicePlane, users, and other CapabilityPlanes for the purpose of requesting services.
- **ApplicationPlane Status (app-status)**
This interface message allows for other CapabilityPlanes and processes to retrieve ApplicationPlane state information.

Figure 9 ApplicationPlane Functions and External Interfaces

Layer Unique Considerations

The ApplicationPlane is intentionally detached from network layer considerations. The purpose of this is to allow the ApplicationPlane to define network service requirements (such as jitter, latency, and bandwidth) and not layer specific technologies.

Security Considerations

The security considerations of the ApplicationPlane is intimately tied to the purpose and intended use of the application, and as such, is beyond the scope of this document.

4 ServicePlane

The ServicePlane refers to the set of systems and processes that are geared towards providing services to users and maintaining state on those services. The ServicePlane will generally rely on the functions of the ControlPlane and/or ManagementPlane to effect actual changes on the DataPlane. In addition, the ServicePlane will typically maintain databases on current and future service instantiations. The ServicePlane also has a very important role in the coordination of other CapabilityPlanes' actions to achieve a higher level goal. This higher level goal will often be initiated by an ApplicationPlane request.

The interface to the ServicePlane is the network demarcation where detailed and specific network service interfaces are defined.

Functions

The following functions are identified for the ServicePlane:

- **Service Management**

This involves processing service requests and subsequently coordinating with the other CapabilityPlanes to service the requests. For inter-domain or inter-technology region actions, the ServicePlane may also coordinate directly with other peer ServicePlanes. The specific services available will be unique to each network. The Service Management function will generally initiate the workflow management processes as described below.

This function also maintains a description of the services offered by a specific instance of a ServicePlane. For instance, a simple service might be a point to point circuit connecting two endpoints. A more complicated service may use multipoint topology to create a broadcast domain between multiple endpoints. Additional features may also be associated with a service offering like the ability to specify latency or jitter requirements. The specifics of the service description will likely be described via an industry standard set of mechanisms, such as web service based XML (Extensible Markup Language) constructs. The service descriptions will be made available to other CapabilityPlanes via the Service Description interface message described below.

- **Workflow Management**

This function will be instantiated in many different forms based on unique service and network requirements. Workflow management refers to the coordination of the functions across the multiple CapabilityPlanes to accomplish a specific set of functions associated with a service provision.

- **ServicePlane Status**

This function involves the maintenance of the ServicePlane state, which support recovery mechanisms that can restore operation after ServicePlane failures. The recovery functions may be completely internal to the ServicePlane, or in some instances may include a coordinated effort across multiple CapabilityPlanes. In this instance, the ServicePlane may be coordinating a larger recovery workflow process. For example, if there was an unrecoverable link failure in a downstream network that caused an end-to-end VC to fail, it would be the responsibility of the ServicePlane to notify the appropriate parties, determine an alternative route if possible, and coordinate the setup of the new VC.

Function Interfaces

The following ServicePlane Functional Interfaces are defined:

- Service Description (serv-descr)
This interface provides a mechanism for a specific ServicePlane to describe the offered services. It is envisioned that other CapabilityPlanes, such as the ApplicationPlane or other ServicePlanes, will utilize this interface to determine service options.
- Service Request (serv-req)
This interface provides for the messaging exchange between the ServicePlane, users, and other CapabilityPlanes for the purpose of requesting services.
- ServicePlane Status (serv-status)
This interface allows for other CapabilityPlanes and processes to retrieve ServicePlane state information.

Figure 10 ServicePlane Functions and External Interfaces

Layer Unique Considerations

The architectural concept of the CapabilityPlane requires a separate and distinct CapabilityPlane for each of the layers. In implementation practice this may not be the case, as a single instantiation of a CapabilityPlane may include capability and functions that cover multiple layers. However for architectural clarity we will maintain this layer separation. A key role of the ServicePlane will be to provide descriptions of the available services and process associated requests. At the ServicePlane layer, there will be two broad categories of services:

- Layer Independent Services - services that are independent of the technology type or layer utilized to provide the service, and
- Layer Dependent Services - services which are dependent on the layer utilized

There is a general desire to define and describe services such that they are Layer Independent Services whenever possible. This is because often a provided "service" is distinct and separate from the type of DataPlane technology utilized. As an example, all DataPlane Types (layer 3, layer 2, layer 1) may include an edge (client access) port as an Ethernet Type. As such, all of these DataPlane types can provide an "Ethernet VLAN Transport Service". For this reason it makes sense to define an "Ethernet VLAN Transport Service" which is layer independent. A service description of this type of service may include parameters such as the following:

"Ethernet VLAN Transport Service" definition in "Parameter:Value" syntax:

- Physical Access Port: *Ethernet*
- Bandwidth: *1 Gigabit/s*

- Ethernet Technology: *802.1Q*
- VLAN Configuration: *VLAN Tag value of 1000*
- QoS Parameters: *<Policing, Burst Specification>*
- Jitter Specification: *<Max Jitter Specification>*
- Latency Specification: *<Max Latency Specification>*

The "Value" fields, shown in italics, are placeholders for specific values which would be acceptable in accordance with the service definition. The detailed service definition will define a range of values which are allowed for each parameter. A requester would then utilize the service definition range to construct a request with specific Parameter:Value combinations.

This type of service could be provided by all of the DataPlane Layers. For instance, it is common today to see this type of service provided over Layer 2.5 (MPLS), Layer 2 (Ethernet), Layer 1.5 (SONET), and Layer 1 (WDM) network types. A similar type of Layer Independent Service could be defined for a "SONET Transport Service" or an "Infiniband Transport Service" or any other technology which can be encapsulated by the various technology layers.

The key decision point as it relates to Layer Independent Services and layers, will be in deciding which layer to use to provide the service. For instance, if an Ethernet Transport Service is requested with a very tight jitter and latency specification, this may motivate the provision of that service over a Layer 1 network as opposed to a higher layer technology. The goal of the ServicePlane is to abstract technology details (as much as possible) from the user, but determine an appropriate layer (i.e. technology) transport that can meet the user's service request.

There will also be Layer Dependent Services. These types of services include specifications which are unique to the Layer and associated DataPlane technology. An example of this type of service would be a "Wavelength Service", where the demarcation between the client (connector) and network is defined in terms of an optical channel. A service description of this type may include parameters such as the following:

"Wavelength Service" Definition in Parameter:Value syntax:

- Physical Access Port: *Optical*
- Wavelength: *1552*
- Power Budget: *<Power Input>*
- Dispersion Specification: *<Dispersion Value>*
- Regeneration Requirements: *<Pure Optical Transport or Regeneration Requirements>*

Another example of a Layer Dependent Service may be a "SONET Switched Service". The "switched" discriminator here as opposed to the "transport" term used earlier in the layer independent services, implies that this service requires layer dependent features. In this case those features are those associated with SONET switched infrastructure and the SONET protocol. A service description of this type may include parameters such as the following:

"SONET Switched Service" Definition in Parameter:Value syntax:

- Physical Access Port: *SONET OC192*
- Bandwidth: *5 Gigabit/s*
- SONET Technologies: *LCAS, VCAT*
- Jitter Specification: *<Max Jitter Value>*
- Latency Specification: *<Max Latency Value>*

This type of service could only be provided for by a Layer 1.5, or a SONET infrastructure. Similar types of Layer Dependent Services could be defined for a "Ethernet Provider Backbone Bridging Traffic Engineering (PBB-TE) Service" or a "Layer 3 Routed Service".

Security Considerations

The security considerations for the ServicePlane will encompass the protection of the ServicePlane system (hardware and software) as well as the network interfaces. The protection at the system level will need to include standard mechanisms for physical security and secure operating system set up and configuration. The security mechanisms for network interfaces will vary depending on deployment environments. It is expected that typical mechanisms will employed such as IPSec[2], firewalls, and access control lists.

5 AAPlane

This plane is the Authentication and Authorization plane. The authentication functions authenticate and identify users. The authorization functions evaluate authorization policy information to determine what actions a user is permitted

Functions

The following functions are identified for the ServicePlane:

- **AA Management**

Authentication and authorization are normally policy-based. For example, authentication may use a table of authorized users with associated passwords, or public keys and possibly attributes such as roles or project membership. Authorization requires some sort of specification of what actions are allowed for users or holders of attributes. It is a function of the AAPlane to define and manage this data. The administration of such information is tightly coupled to the mechanisms that underpin the security paradigms used by the various capability planes. For instance, in a non-federated framework, the management of AA information may be local to the service provider. In a federated style AA framework, such as a virtual organization (VO), management of AA information may be widely distributed. In the later case, there may be additional requirements for AA methods between the components within the trusted VO boundary, for example, between the identity provider (IDP) and attribute server. Authorization policy consists of information such as identities, attributes, resources, and rights, and the relationships between them that is used to determine what actions an authenticated user is allowed to perform. In the context of the ManagementPlane, it may define role attributes such as network operators and network engineers, the latter, e.g., having a superset of rights (or permissions) to change network parameters. In the ControlPlane, it may be a list of permitted neighbors that control plane signaling messages can be accepted from. For the ServicePlane, the authorization policy information can embody the Acceptable Use Policy (AUP) of the service provider.

- **AA AuthN**

This is the function that identifies the user. It takes some type of user credential, such as a username and password, or signed message and key identifier or identity provider handle. It verifies the credential and returns the local identifier and possibly attributes for the user. Information needed to authenticate a user may be retrieved from the authentication policy, e.g., if a username/password or public key is required or from external sources such as IDPs, e.g., if Shibboleth authentication is required. The AA AuthN function conceptually encompasses the various authentication methods that are necessary for each of the distinct capability planes. For example, if a private/public key authentication scheme is used by network operators to access the ManagementPlane, this function must determine the validity of the private key based on the authentication policy specified key store containing the corresponding public key certificate, or the trusted certificate authority that issued it. If the ControlPlane requires the use of cryptographic authentication (e.g. MD5) for control plane signaling messages, this function must decrypt the message and verify its validity. Other common authentication schemes include mechanisms such as one-time passwords, user attribute verification (e.g. Shibboleth), and credential validation (e.g. RADIUS).

- **AA AuthZ**

This is the function that verifies the right of the user to perform the requested action. It takes an authenticated user identity and attributes and the requested action and applies the authorization policy. This essentially embodies the policy decision point (PDP). For the ManagementPlane, the AA AuthZ could verify if an SNMP MIB could be read or written to, or which parts of the network elements could be viewed or edited. In the ControlPlane, it could determine if the bandwidth request in a control plane signaling message is valid – that is, within the range permitted to the user by policy. Within the ServicePlane, the AA AuthZ decision could be as simple as determining if a user's request is within a certain limit, or as complex as whether it is

within the amount of bandwidth that can be requested for a specific link at a specific time of the day (e.g. on-peak/off-peak times).

- **AAPlane Status**

This function involves the maintenance of AAPlane state such that recovery mechanisms can restore operation after AAPlane failures such as corruption of the authentication or authorization policies. The recovery functions may be completely internal to the AAPlane, or in some instances may include a coordinated effort across multiple CapabilityPlanes. In this instance, the ApplicationPlane may be providing state information as part of a larger recovery workflow process. Mechanisms might include automatic backups of the policy and keeping an audit trail of any changes to policy.

Function Interfaces

The following AAPlane Functional Interfaces are defined:

- **AA AuthN (aa-authn)**

This interface is used by the various capability planes to authenticate the user. This interface also serves as a secure connection to external AA entities such as IDPs and attribute servers in the case of federated services.

- **AA AuthZ (aa-authz)**

This interface is used to determine the access privilege of the user. In the case of inter-domain authorizations, the access privileges may be determined based on the authenticated identity of the requesting domain and not the individual initiating the request. This paradigm would occur, for example, if a service level agreement (SLA) existed between the networks of two peering administrative domains with respect to the maximum amount of priority traffic that is allowed to pass from one domain to another.

- **AA State (aa-status)**

This interface message allows for other CapabilityPlanes and processes to retrieve AAPlane state information.

Figure 11 AAPlane Functions and External Interfaces

Layer Unique Considerations

The AAPlane does not directly interact with the DataPlane, and as such is network layer agnostic. Access to, and manipulation of the DataPlane is abstracted from the AAPlane by the ManagementPlane and ControlPlane.

Security Considerations

The security considerations for the AAPlane mainly encompass the protection and manipulation of the AuthN and AuthZ information that constitute the PDP. This could range from physical security of the system running the AAPlane, to encryption of the PDP information.

6 ControlPlane

The ControlPlane plane is responsible for routing, path computation, and signaling functions associated with control of the DataPlane. This generally includes maintaining topology information and directing network elements in terms of data ingress, egress, and switching operations.

For all the CapabilityPlanes the collection of devices over which authority is exercised is a single network “technology region” as described in Section 1. The ControlPlane is one of two CapabilityPlanes which directly interacts with the DataPlane. As a result there will be differences in the data collected and maintained by the ControlPlane depending on the type of network region under control. For instance a DataPlane based on a L2SC technology region may require the ControlPlane to manage the Ethernet Virtual Local Area Network (VLAN) tag space, while a DataPlane based on a LSC technology region may require the ControlPlane to manage wavelength based resources. These types of technology specific distinctions are noted and described in the DataPlane section. In terms of the ControlPlane Functions described in this section, the functions and interfaces here are at a level such that they apply to all DataPlane technology types.

Functions

The following functions are identified for the ControlPlane:

- **Routing**
Routing protocols are responsible for the reliable advertisement of the network topology and the available resources (such as bandwidth and other technology specific features) within the network region. This function provides the distribution and dissemination of reachability information, layer and technology specific information, resource usage status, and topology information, between network elements within network region. At a given DataPlane technology region, the routing information may be either distributed or centralized. In multi-layer, multi-technology, multi-vendor, multi-domain environments, this function may be realized using either pre-configured information (i.e., static topology input) or dynamic state (inter-domain routing) between domain-level routing entities.
- **Path Computation**
This function refers to the processing of routing information to determine how to accomplish functions such as provisioning an end-to-end path, evaluating network state, adjusting to failures/changes, or instantiating constraint-based topologies. In a multi-layer, multi-technology, multi-vendor, multi-domain environment, this may require many specific discrete functions such as traffic-engineering database pruning, network graph construction and transformations, and multi-dimensional constrained shortest path first (CSPF), and possibly additional algorithms.
- **Signaling**
Signaling protocols are responsible for provisioning, maintaining, and deleting connections and the exchange of messages to instantiate specific provisioning requests based upon the above routing and path computation functions. This may be accomplished via protocols such as RSVP-TE or Web-Service-based messaging, for example. In multi-layer, multi-technology, multi-vendor, multi-domain environments, this function will likely require appropriate translation at the signaling interfaces between respective control plane entities.
- **Traffic Engineering DataBase (TEDB)**
This is the function inside the control plane which stores the topology state of the DataPlane. The information in the TEDB will come from the routing information as well as from external sources such as the other capability planes.
- **ControlPlane Status**

This function involves the maintenance of ControlPlane state such that recovery mechanisms can restore operation after ControlPlane failures. The recovery functions may be completely internal to the ControlPlane, or in some instances may include a coordinated effort across multiple CapabilityPlanes. In this instance, the ControlPlane may be providing state information as part of a larger recovery workflow process. As an example, a ManagementPlane process may be investigating a problem with a specific circuit. This may require the ManagementPlane to first obtain some ControlPlane status, such as circuit identifiers, which it will then utilize to directly query the DataPlane to obtain further information.

Function Interfaces

The following ControlPlane Functional Interfaces are defined:

- Routing Data (ctrl-rd)
This is the interface between the ControlPlane and the DataPlane for the purpose of obtaining routing and topology data.
- Path Computation (ctrl-pc)
This is the interface which allows for path computation request and replies.
- Signaling Message (ctrl-sig)
This is the interface which initiates provisioning actions on the DataPlane.
- ControlPlane Status (ctrl-status)
This interface message allows for other CapabilityPlanes and processes to retrieve ControlPlane state information.

Figure 12 ControlPlane Functions and External Interfaces

Layer Unique Considerations

The ControlPlane is one of two CapabilityPlanes which directly touch the DataPlane. Therefore there will be many layer unique considerations for a ControlPlane in terms of what type of DataPlane it is managing. For instance a ControlPlane which is managing a Layer 3 Packet Switch Capable (PSC) DataPlane type will control the technologies specific to that layer which include IP, packets, QoS (Quality of Service), bandwidth management, and forwarding tables. A ControlPlane which is covering a Layer 1 Lambda Switch Capable (LSC) DataPlane type will control technologies specific to that layer which include items such as wavelength routing, conversion, and switch states.

There will be set of layer unique considerations for each of the individual DataPlane types. The specifics of these DataPlane types are presented in Section 8 and 9. The ControlPlane will be required to use the Function and Function Interfaces as described for each DataPlane type.

Security Considerations

The security considerations for the ControlPlane will encompass the protection of the system (hardware and software) as well as the network interfaces. The protection at the system level will need to include standard mechanisms for physical security and secure operating system set up and configuration. The security mechanisms for network interfaces will vary depending on deployment environments. It is expected that typical mechanisms employed to create secure conduits for ControlPlane signaling or control messages would include IPSec[2], firewalls, and access lists. This is to prevent unauthorized access from ControlPlanes in other network domains.

In addition, the ControlPlane may employ protocol specific mechanisms to enhance security. The specifics of these types of mechanisms will vary based on individual deployments and considered in the context of all total security considerations. Typical examples of protocol specific security mechanism may include technologies like BGP MD5 [3], OSPF Authentication[4], or new research project developed solutions.

7 ManagementPlane

The ManagementPlane refers to the set of systems and processes that are utilized to monitor, manage, and troubleshoot the network. This includes functions to support user accessible services as well as network maintenance, upgrades, and reconfigurations. This plane is responsible for collecting data and monitoring of the network. In addition, this CapabilityPlane may also include capabilities to configure the network elements with the support of the control plane or via independent actions.

The ManagementPlane is one of two CapabilityPlanes which directly interacts with the DataPlane. As a result there will be differences in the data collected and maintained by the ManagementPlane depending on the type of network region under control. For instance a DataPlane based on a L2SC technology region may require the ManagementPlane to manage the state of Ethernet Virtual Local Area Network (VLAN) configurations on Ethernet switches. While a DataPlane based on a LSC technology region may require the ManagementPlane to manage the state of wavelength cross connects on a lambda switch. These types of technology specific distinctions are noted and described in the DataPlane section. Monitoring functions may be accomplished using SNMP (Simple Network Management Protocol), CLIs (Command Line Interfaces), TL1 (Transaction Language 1), or other DataPlane specific mechanisms. In terms of the ManagementPlane Functions described in this section, the functions and interfaces here are at a level such that they apply to all DataPlane technology types.

The ManagementPlane may be queried by network administrators, users and other capability planes such as the ServicePlane or the ApplicationPlane. The ManagementPlane may publish monitoring data, or meta data, which is available for other domains to access.

The ManagementPlane and ControlPlane interoperation is worthy of special note. As the two CapabilityPlanes that directly interact with the DataPlane, detailed planning is required in terms of their interaction with the DataPlane for network element configurations and provisioning. Technically, the ManagementPlane and ControlPlane are functionally distinct, with operations that are independent and autonomous. However, in practice, there will be tight coordination and in many cases cooperation between the two. An example of coordination may be the partitioning of DataPlane resources into ControlPlane provisioned and ManagementPlane provisioned components. An example of this might be the end-to-end setup of an Ethernet VLAN service, whereby some of the devices in the path support Ethernet over an MPLS LSP which is instantiated by the ControlPlane, and some devices are “dumb” switches which are configured to bridge Ethernet VLANs via the ManagementPlane. An example of cooperation would be the ManagementPlane invoking desired network element service provisioning via the ControlPlane mechanisms as opposed to independent actions. For instance, the configuration of an MPLS LSP on a router is done via the ManagementPlane, however the signaling and instantiation of the LSP is done by the ControlPlane. Another example of cooperation would be automatic status notifications exchanged between the ControlPlane and ManagementPlane. A typical example of this is the ControlPlane automatically notifying the ManagementPlane

regarding service provisioning actions, and the ManagementPlane notifying the ControlPlane regarding network element or link failures.

Functions

The following functions are identified for the ManagementPlane:

- **Monitoring**
This function involves querying each of the other capability planes for information. This includes information such as:
 - DataPlane - network element status, utilization information, and resource configuration, interface information (errors, utilization, MTU configurations)
 - ControlPlane - status and information
 - ServicePlane - status and information
 - AAPlane - status and information
 - ManagementPlane - components of the ManagementPlane may be responsible for monitoring other ManagementPlane components as well
- **Troubleshoot Procedures**
This involves the investigation of issues and problems in the network. The troubleshooting procedures are generally a set of monitoring steps conducted in an objective specific manner to identify or resolve an issue. The issues are generally identified by a user, another CapabilityPlane, or the ManagementPlane itself.
- **ManagementPlane Status**
This function involves the maintenance of ManagementPlane state such that recovery mechanisms can restore operation after ManagementPlane failures. The recovery functions may be completely internal to the ManagementPlane, or in some instances may include a coordinated effort across multiple ManagementPlanes. In this instance, the ManagementPlane may be providing state information as part of a larger recovery workflow process.

The following ManagementPlane Functional Interfaces are defined:

- **Monitoring (man-mon)**
This interface is the messaging exchanged between the ManagementPlane and the DataPlane to obtain network element status, utilization information, and resource configuration information.
- **Troubleshoot (man-trbl)**
This interface is the messaging exchanged between the ManagementPlane and other CapabilityPlanes (or users) to initiate and respond to troubleshooting requests.
- **Recovery (man-status)**
This interface message allows for other CapabilityPlanes and processes to retrieve ManagementPlane state information.

Layer Unique Considerations

The ManagementPlane must recognize many layer unique considerations in terms of what type of DataPlane it is covering. In general, each of the various DataPlane “technology regions” will expose information that details the health of the data plane. However the information for each technology region will be technology specific. For example, monitoring a VLAN would involve looking at the state of the spanning tree protocol, but monitoring an MPLS LSP would require looking at the OSPF-TE database and RSVP messages.

Figure 13 ManagementPlane Functions and Interfaces

Here are some examples of monitoring data that can to be exposed by the ManagementPlane:

- Layer 3 for PSC using IP Routing: packet drops, CRC errors, unicast/multicast packet counts
- Layer 2.5 for PSC using MPLS: LSP status, RSVP neighbor adjacencies
- Layer 2 for L2SC: link errors, bit error rate (BER),
- Layer 1.5 for TDM: BER, SONET Circuit Status and Error Codes
- Layer 1 for LSC: optical power monitoring, WDM Optical Channel Status and Error Codes

Monitoring data may be collected either by a push model, such as SNMP traps or Syslog messages, or a pull model such as SNMP polling, TL1, or SCP. This data can subsequently be published via a ManagementPlane API.

The specifics of these DataPlane types are presented in Section 8 and 9. The ManagementPlane will be required to monitor and manage many items related to the Function and Function Interfaces as described for each DataPlane type.

Security Considerations

The security considerations for the ManagementPlane is mainly focused on preventing unauthorized access to the ManagementPlane itself. In addition to authentication and authorization methods (as described in the AAPlane section), isolating access or connectivity to the ManagementPlane is another common practice. This can be done by using a completely isolated (i.e. out-of-band) network or an in-band Virtual Private Network (VPN).

In addition, the ManagementPlane may employ protocol specific mechanisms to enhance security. The specifics of these types of mechanisms will vary based on individual deployments and considered in the context of all total security considerations. Typical examples of protocol specific security mechanism may include technologies like SNMPv3 [5], CLI based authentication, or new research project develop solutions.

8 DataPlane

The DataPlane is the set of network elements which receives, sends, and switches the network data. For this architecture definition we identify the dataplane options in terms of "technology regions". A "technology region" is a set of network elements which are grouped together and utilize the same dataplane "technology type". The technology types are defined using the standard Generalized Multi-Protocol Label Switching (GMPLS) [1] nomenclature of Packet Switching Capable (PSC) layer, Layer-2 Switching Capable (L2SC) layer, Time Division Multiplexing (TDM) layer, Lambda Switching Capable (LSC) layer, and Fiber-Switch Capable (FSC). Additionally, we associate these technology types with the more common terminology as follows:

- Layer 3 for PSC using IP Routing
- Layer 2.5 for PSC using MPLS
- Layer 2 for L2SC (often Ethernet)
- Layer 1.5 for TDM (often SONET/SDH)
- Layer 1 for LSC (often WDM switch elements)
- Layer 0 for FSC (often port switching devices based on optical or mechanical technologies)

There are unique features and capabilities associated with each of the technology types. These are discussed in Section 9. In addition, the networks of most interest will be constructed by utilizing a combination of two or more of these DataPlane technologies to form a multi-layer hybrid network. The features, benefits, and challenges of multi-layer networks are discussed in Section 10.

There are multiple implementation options for all of the technology types. This includes IP Routers (with MPLS capabilities), Transport-MPLS (T-MPLS), Ethernet, Ethernet Provider Backbone Bridge Traffic Engineering (PBB-TE), Synchronous optical networking (SONET)/Synchronous Digital Hierarchy (SDH), next-generation SONET/SDH, Wave-Division Multiplexing (WDM), and Micro-Electro-Mechanical Systems (MEMS) or mechanical based port switches. The details of specific technology type implementations are not discussed in this document.

In this section we define the base functions and interfaces for the DataPlane. Initially, we define the set of functions and interfaces at a level where they are generic across all the DataPlane types. The sub-sections then discuss the technology specific features of the different technology types.

Functions

The following functions are identified for the DataPlane:

- **Element Control**
This refers to the physical configuration of the network elements in the dataplane. The specific control mechanism will be unique to the DataPlane technology and capabilities.
- **Element Status**
This refers to obtaining and providing information on the status of network elements in the dataplane. Typically the ManagementPlane will be the primary consumer of information from this DataPlane function.
- **Layer Adaption**
This refers to the DataPlane adaptation from one technology type to another. The specific adaptation capabilities will be unique to the DataPlane technology and capabilities. A common adaptation type is that accomplished by DataPlane elements which have Ethernet client ports which are adapted into SONET/SDH or WDM for transmission over wide area links.

Interfaces

The following DataPlane Functional Interfaces are defined:

- **Control (dp-ctrl)**
This interface provides the messaging to allow for DataPlane network elements to be configured. The ControlPlane is typically the user of this interface.
- **Status (dp-status)**
This interface provides the messaging to allow for DataPlane network elements to be monitored and status obtained. The ManagementPlane is typically the user of this interface.
- **Layer Adaptation (dp-adapt)**

This interface allows for the layer adaptation features to be controlled and configured. This typically includes features to select from available adaptation options like encapsulation technologies or encoding parameters.

Figure 14 DataPlane Functions and External Interfaces

Layer Unique Features

The DataPlane types as described earlier in this section define the layer unique categories that the other CapabilityPlanes must consider. As a result, the DataPlane does not have layer unique considerations as described for the other CapabilityPlanes, but instead has layer unique features. These layer unique features are as described in the Section 9, Specific DataPlane Types.

Security Considerations

The security considerations for the DataPlane is mainly focused on preventing unauthorized access, disruption, or passive monitoring of the DataPlane itself. This is typically accomplished by a combination of facility security mechanisms to prevent physical access to network elements, as well as protection of the system interfaces from a computer system access perspective. In addition, the DataPlane may have embedded security features which can be exercised by the ControlPlane of ManagementPlane. An example of this would be a hardware based encryptor which may be tightly coupled into a component of the DataPlane. However, these types of things will be unique to individual DataPlane implementations, and for now we simply mention the possibility of features such as this. In general the same security feature could be implemented at different CapabilityPlane levels. The decision regarding where to put many security features will be driven more by overall system architecture and implementation details as opposed to anything unique about the various CapabilityPlanes or DataPlane layer types.

9 Specific DataPlane Types

The previous architecture description has described a generic DataPlane and the associated functions. In the following section we identify the specific DataPlane types and discuss the unique features and capabilities associated with each.

The technology types that follow are based on RFC 3945, Generalized Multi-Protocol Label Switching (GMPLS) Architecture [1] nomenclature of Packet Switching Capable (PSC) layer, Layer-2 Switching Capable (L2SC) layer,

Time Division Multiplexing (TDM) layer, Lambda Switching Capable (LSC) layer, and Fiber-Switch Capable (FSC). Additionally, we associate these technology types with the more common terminology as follows:

- Layer 3 for PSC using IP Routing
- Layer 2.5 for PSC using MPLS
- Layer 2 for L2SC (often Ethernet)
- Layer 1.5 for TDM (often SONET/SDH)
- Layer 1 for LSC (often WDM switch elements)
- Layer 0 for FSC (often port switching devices based on optical or mechanical technologies)

Table 1 below identifies the icons which are utilized for each of these technology types.

Table 1 DataPlane Types and Icons

DataPlane Technology Type	Common Layer Terminology	Icon
Packet Switching Capable (PSC)	Layer 3
Packet Switching Capable (PSC) with MPLS	Layer 3/Layer 2.5
Layer-2 Switching Capable (L2SC)	Layer 2
Time Division Multiplexing (TDM)	Layer 1.5
Lambda Switching Capable (LSC)	Layer 1
Fiber-Switch Capable (FSC)	Layer 0

The DataPlane type specific features are identified as they relate to the ControlPlane and ManagementPlane functions.

9.1 PSC

Packet Switch Capable (PSC) technology regions have elements that recognize packet boundaries and can forward data based on the content of the packet header. Examples include interfaces on routers that forward data based on the content of the IP header and interfaces on routers that switch data based on the content of the MPLS "shim" header.

Figure 15 PSC DataPlane Type

Figure 16 PSC with MPLS DataPlane Type

9.2 L2SC

Layer-2 Switch Capable (L2SC) technology regions have elements that recognize frame/cell boundaries and can switch data based on the content of the frame/cell header. Examples include interfaces on Ethernet bridges that switch data based on the content of the MAC header and interfaces on ATM-LSRs that forward data based on the ATM VPI/VCI.

Figure 17 L2SC DataPlane Type

9.3 TDM

Time-Division Multiplex Capable (TDM) technology regions have elements that switch data based on the data's time slot in a repeating cycle. An example of such an interface is that of a SONET/SDH Cross-Connect (XC), Terminal Multiplexer (TM), or Add-Drop Multiplexer (ADM). Other examples include interfaces providing G.709 TDM capabilities (the "digital wrapper") interfaces.

Figure 18 TDM DataPlane Type

9.4 LSC

Lambda Switch Capable (LSC) technology regions have elements that switch data based on the wavelength on which the data is received. An example of such an interface is that of a Photonic Cross-Connect (PXC) or Optical Cross-Connect (OXC) that can operate at the level of an individual wavelength. Additional examples include PXC interfaces that can operate at the level of a group of wavelengths, i.e., a waveband and G.709 interfaces providing optical transport capabilities.

Figure 19 LSC DataPlane Type

9.5 FSC

Fiber-Switch Capable (FSC) technology regions have elements that switch data based on a position of the data in the (real world) physical spaces. An example of such an interface is that of a PXC or OXC that can operate at the level of a single or multiple fibers.

Figure 20 FSC DataPlane Type

10 Multi-Layer Architectures

In this section we review the consideration of building networks which have two or more technology regions, along with the associated CapabilityPlanes. There are multiple possible configurations for such networks and we classify them into the following categories:

- Multi-Layer Networking - Vertical
- Multi-Layer Networking – Horizontal
- Multi -Layer Networking – Combined
- Multi-Layer Networking – InterDomain

The following sections describe each of these primarily from a DataPlane topology and connection perspective. In addition, there will be interactions between the associated CapabilityPlanes as well. These CapabilityPlane interactions can be quite varied depending on the system requirements and purposes. The workflow descriptions in Section 2 are intended to capture several example CapabilityPlane messaging and service provisioning scenarios. The descriptions of multi-layer and multi-region networking described here is similar to that described in Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN), RFC 5212 [6].

10.1 Multi-Layer Networking - Vertical

In this section we discuss a multi-layer network topology where two or more DataPlane types may be layered in a vertical manner. The notion of a vertical layering of technology regions implies using lower layer service provisioning to provide capabilities at the higher layers. Figure 21 depicts a vertical multi-layer topology consisting of PSC, L2SC, and LSC technology regions. As noted in the figure, each of the technology regions has its own set of CapabilityPlanes. In addition, technology adaptation points are required in order to move data across layer boundaries.

Figure 21 Multi-Layer Networking - Vertical

A typical provisioning action for a vertical multi-layer topology such as this would be for a lower layer, such as the LSC region, to provision a circuit which would be reflected as a link at the higher L2SC layer. Subsequently L2SC services may be provided. A similar scenario could be accomplished using the L2SC as the lower layer and the PSC region as the higher layer.

10.2 Multi-Layer Networking - Horizontal

In this section we discuss a multi-layer network topology where two or more DataPlane types may be layered in a horizontal manner. Figure 22 depicts the notion of a horizontal layering of technology regions. This topology implies that we are integrating or “stitching” services across technology region boundaries. Different from the vertical case, we do not use a lower layer service to create a link or capability at a higher layer.

A typical provisioning action for a horizontal multi-layer topology such as this would be to provision a path across multiple technology regions in order to provide a service. This service will generally present a similar technology (like Ethernet) at both edges, but the underlying technology may differ along the path.

As noted in the figure, each of the technology regions has its own set of CapabilityPlanes. In addition, technology adaptation points are required in order to move data across technology region boundaries.

Figure 22 Multi-Layer Networking - Horizontal

10.3 Multi-Layer Networking - Combined

We can combine the vertical and horizontal multi-layer topologies to create a more flexible and sophisticated set of available network services. As shown in Figure 23, a topology may consist of vertical multi-layer networks peering in a horizontal manner. In this context, all the peering links which cross the boundary line represent horizontal multi-layer networking. There are two of these types of links shown in Figure 23, but there could be more.

Figure 23 Multi-Layer Networking - Combined

It should be noted that the horizontal peering links at the higher layer (Layer 3 in this example) may be built via a direct physical link between the two Layer 3 devices, or may have been created via an earlier provisioning of some resources from the horizontal lower layer (Layer 1 in this example) link. In either case the architecture and subsequent handling of provisioning events will be identical. There will be some practical impacts associated with provisioning of higher layer links from lower layer links, such as reduced available capacity on the lower layer link for future provisioning actions. However, the ability to provision services at both layers independently or in an integrated fashion remains.

10.4 Multi-Layer Networking - InterDomain

The notion of InterDomain messaging and service provisioning is similar to the horizontal multi-layer networking case, where the horizontal boundary line is also a domain boundary. The primary difference between the two will be a matter of tailoring the full set of inter-region communications to a subset that will meet the security and scalability requirement for InterDomain communications. The service workflows in the Section 2 capture some of these notions to further clarify this point.

10.5 Hybrid Networking (Multi-Layer Traffic Engineering and Traffic Grooming)

The concept of Hybrid Networking is introduced here in the context of using multi-layer networks to conduct sophisticated management and movement of data flows across the various layers in a flexible and dynamic fashion.

The intelligence and processes required to determine "why and when" to perform such functions is beyond the scope of this architecture document. This is a topic that is envisioned be a research, development, deployment subject of increasing importance as multi-layer network architecture are increasingly deployed.

This architecture document does cover the "how" with respect to Hybrid Networking functions. Specifically, the ServicePlane interface will provide an entry point into one or all of the DataPlane layers with where a Hybrid Networking agent could obtain network service and topology provisioning to support larger Hybrid Networking workflows. In general, it is anticipated that such a Hybrid Networking agent would utilize the, Multi-Layer Networking (Vertical, Horizontal, Combined, InterDomain) capabilities as needed and as available to accomplish its larger goals of Hybrid Networking traffic engineering and traffic grooming. In this context at Hybrid Networking Agent would look like a process operating at the ApplicationPlane from the Multi-Layer Network architecture perspective.

10.6 Nested Capability Planes

The concept of nested CapabilityPlanes is another topic of interest. An action is classified as Nested when the resulting network services (or topologies) are handed off to a separate set of CapabilityPlanes for subsequent responsibility. An action where the resulting network services (or topologies) are not handed off to a separate set of CapabilityPlanes is not considered a nested action.

An important point to note here is that the ServicePlane service interface and feature set does not change for the nested vs non-nested concept. The only distinction between the two is how the requesting entity utilizes the results of a requested service instantiation. The most obvious scenario for the nested case is the provision of an entire topology which is handed off from one ServicePlane to a second ServicePlane (and associated CapabilityPlanes) which then assumes responsibility for subsequent service instantiations.

The Multi-Layer Networking-Vertical type is an example of a Nested Capability Plane action. However, the Nested Capability Plane concept is really broader than that limited example. The Nested CapabilityPlane topic is intended to encompass a larger range of functions and systems which utilizes Multi-Layer Networking Capabilities in sophisticated and complex ways to create and manage multiple "virtual" network topologies. These topologies may appear independent to some CapabilityPlanes but which in reality are created via a recursive hand-off of resource sub-sets from one CapabilityPlane instance to another.

The intelligence and processes required to determine "why and when" to perform such Nested CapabilityPlane functions is beyond the scope of this architecture document. This is a topic that is envisioned be a research, development, deployment subject of increasing importance as multi-layer network architecture are increasingly deployed.

This architecture document does cover the "how" with respect to Nested CapabilityPlane processes. Specifically, the ServicePlane interface will provide an entry point into the DataPlane layers where a Nested CapabilityPlane agent could obtain network service and topology provisioning to support larger Nested CapabilityPlane workflows. In general, it is anticipated that such a Nested CapabilityPlane agent would utilize the Multi-Layer Networking

(Vertical, Horizontal, Combined, InterDomain) capabilities as needed and as available to accomplish its larger goals of Nested CapabilityPlane topology instantiations.

11 Glossary

Multi-Layer Network - A network which consists of two or more technology or DataPlane layers.

CapabilityPlanes - generic name for a set of functions and processes which are responsible for a specific functional area in a multi-layer network. The following CapabilityPlanes are identified: DataPlane, ControlPlane, ManagementPlane, AAPlane, ServicePlane, ApplicationPlane.

DataPlane - A set of network elements which receive, send, and switch the network data. The following types of DataPlanes are identified in this architecture document:

- Layer 3 for PSC using IP Routing
- Layer 2.5 for PSC using MPLS
- Layer 2 for L2SC (often Ethernet)
- Layer 1.5 for TDM (often SONET/SDH)
- Layer 1 for LSC (often WDM switch elements)
- Layer 0 for FSC (often port switching devices based on optical or mechanical technologies)

ControlPlane - The set of functions and processes responsible for routing, path computation, and signaling functions associated with control of the DataPlane.

ManagementPlane - The set of functions and processes that are utilized to monitor, manage, and troubleshoot the network.

AAPlane - The set of functions and processes that are provided to allow the other planes to identify and authenticate users and receive associated policy information.

ServicePlane - The set of functions and processes that are geared towards providing services to users and maintaining state on those services.

ApplicationPlane - The set of higher level functions which will generally be tailored for domain specific purposes. The ApplicationPlane will rely on the capabilities offered by one or more ServicePlanes to accomplish its objectives.

Service Workflow - A coordinated set of actions involving multiple CapabilityPlanes to accomplish a specific layer network function.

Packet Switch Capable (PSC) technology regions have elements that recognize packet boundaries and can forward data based on the content of the packet header. Examples include interfaces on routers that forward data based on the content of the IP header and interfaces on routers that switch data based on the content of the MPLS "shim" header.

Layer-2 Switch Capable (L2SC) - DataPlane type which has network elements that recognize frame/cell boundaries and can switch data based on the content of the frame/cell header.

Time-Division Multiplex Capable (TDM) - DataPlane type which has network elements that switch data based on the data's time slot in a repeating cycle.

Lambda Switch Capable (LSC) - DataPlane type which has network elements that switch data based on the wavelength on which the data is received.

Fiber-Switch Capable (FSC) - DataPlane type which has elements that switch data based on a position of the data in the (real world) physical spaces.

Hybrid Networking - The process of using multi-layer networks to conduct sophisticated management and movement of data flows across the various layers in a flexible and dynamic fashion.

Nested CapabilityPlane - The concept of using the action of one CapabilityPlane to instantiate a network services (or topology which is then handed off to a separate instance of a CapabilityPlane for subsequent responsibility and control.

12 Authors

The authors of this document are as follows:

Nasir Ghani
University of New Mexico (UNM)

Chin Guok
Energy Sciences Network (ESnet)

Tom Lehman
University of Southern California
Information Sciences Institute
(USC/ISI)

Brian Tierney
Energy Sciences Network (ESnet)

13 Acknowledgements

We would like to thank Mary Thompson and William Johnston for their help with this document. We would also like to thank the ASCR program office for their help and advice. This work was supported by the Directors of the Office of Science, Office of Advanced Scientific Computing Research, U. S. Department of Energy under Contract No.DE-AC02-05CH11231 and Contract No.DE-FG02-06ER25741.

Appendix 1 Standards Bodies Overview

The section provides an overview of some key standards bodies activities. The activities discussed here are primarily focused in the Control Plane area. The other CapabilityPlane areas do not have a large and focused standards bodies effort similar to that for the Control Plane.

ITU-T

The ITU-T has been maturing its multi-domain capable ASTN framework for several years (G.8080, formerly G.ASON) [7]. The reference architecture here defines a hierarchical setup of routing areas (RA). At the lowest hierarchical level, a RA represents a domain comprising of physical nodes and links. At higher levels an RA represents multiple “abstract” nodes and links. Note that asynchronous transfer mode (ATM) also defined a hierarchical design with peer-groups, i.e., private network-to-network interface (PNNI) protocol. Now ASTN further defines component groups to setup, maintain, and release client connections, e.g., an RA can have one/more routing controller (RC) entities. Associated component functions are also outlined for tasks such as auto-discovery, auto-provisioning, restoration, etc. In ASTN, network topology is not made visible to the client layer and hence connections are treated as sub-network point pool (SNPP) links. Overall, ASTN is quite flexible as each lower-layer control plane can be tailored to the particular type of equipment supported. Nevertheless, as ASTN only defines architectures, its liaison efforts with the IETF and OIF are of crucial importance.

OIF

The OIF has largely focused on optical interfacing protocols, including a client-network UNI and a network-network NNI [8]. UNI defines bandwidth signaling for client devices (i.e., IP/MPLS routers) to request/release “optical” connections from carrier SONET/SDH or DWDM domains, i.e., “optical dial-tone”. Since there is no trust relationship here, resource/topology state is not propagated to clients, i.e., overlay model. The latest UNI 2.0 features much-improved capabilities for security, bandwidth modification, etc. Meanwhile the NNI implements inter-domain functionality for reachability/resource exchange and setup signaling and features two variants, interior NNI (I-NNI) and external-NNI (E-NNI). The former interfaces nodes within the same administrative area whereas the latter serves adjacent (possibly multi-carrier) areas. Namely, E-NNI relegates all interfacing to domain boundaries, thereby removing restrictions on domain-internal control and equipment interoperability. Recently the OIF has detailed routing and signaling functionalities for E-NNI. Specifically, a hierarchical routing setup is defined (ASTN G.8080) based upon OSPF-TE. However the inter-carrier case has not been fully addressed yet. Overall, UNI and NNI can automate circuit setups across multiple “optical” layers, DWDM and TDM.

IETF

Internet Protocol (IP) has a mature multi-domain setup comprising a hierarchy of autonomous systems (AS) and areas (domains). Within areas, routers run interior gateway protocols (IGP) such as open shortest path first (OSPF) or intermediate-system to intermediate-system (IS-IS) to maintain link state databases (LSDB). Meanwhile, the inter-AS level uses exterior gateway protocols (EGP), most notably distance vector border gateway protocol (BGP), for reachability exchange. However BGP represents a very high level of aggregation and is insufficient for TE circuit routing requiring link/path state. Here OSPF-TE can provide an added routing level for inter-domain operation. With growing quality of service (QoS) needs, OSPF also defines TE extensions (OSPF-TE, RFC 2676) for new opaque link state attributes (LSA). Namely, these entities can disseminate “QoS-related” state to support advanced constraint-based routing (CBR). Some have also tabled QoS destination extensions for BGP.

Furthermore, the IETF has also extended control provisioning to optical domains by augmenting its MPLS suite, i.e., generalized MPLS (GMPLS) [x2]. For routing this includes new OSPF-TE opaque LSA definitions for DWDM and SONET/SDH links, allowing TE databases (TEDB) to store wavelengths/usages, timeslots/usages, shared risk link groups (SRLG), etc. Meanwhile for signaling, extended RSVP-TE now supports hard state circuit setup/takedown, recovery, etc. A new link management protocol (LMP) has also been defined for resource discovery and fault localization. Now it is important to consider the applicability of GMPLS for multi-domain networking. From a routing angle, OSPF-TE can suffice as a unified inter-domain link-state routing protocol as it supports multiple link granularities. Meanwhile RSVP-TE offers many salencies for multi-domain circuit signaling via its loose route (LR) feature [x2]. Namely, partial skeleton routes can be specified and subsequent explicit route (ER) expansion used to resolve full label switched paths (LSP). RSVP-TE also defines mechanisms for LSP setup across domain

boundaries—contiguous, stitched, and nested. Carefully note that proxy devices can also be used to also incorporate proprietary domains under GMPLS control.

Another IETF multi-domain standard is the path computation element (PCE) framework [9], which decouples TE path computation from signaling. In this setup a domain can have one/more logical (standalone or co-located) PCE entities which communicate with path computation clients (PCC) to resolve connection paths. All PCC-PCE communication is done via a new PCE protocol (PCEP). Although a PCE has access to local domain resource/policy databases, its inter-domain visibility may vary. In the simplest case, a PCE may only have knowledge of its domain egress, i.e., local visibility (low-trust, inter-carrier). Alternatively a PCE may have knowledge of physical inter-domain links and even resources in external domains, i.e., partial visibility (high-trust, intra-carrier). Accordingly, two distributed path computation schemes are envisioned, per-domain and PCE-based. The former computes paths in a “domain-domain” manner and is suitable for limited visibility. Namely, PCE entities (or border nodes) iteratively compute TE paths across their domains to ingress nodes in the next domain. Meanwhile the latter operates with increased inter-domain visibility and two strategies have been tabled, multi-PCE path computation with/without inter-PCE signaling. Note that the PCE framework also allows policy control at domain boundaries—a crucial requirement in multi-carrier settings, on par with TE objectives. Specifically, an ingress PCE can enforce policies to determine which requests it will support along with associated TE constraints/algorithms.

MEF

The Metro Ethernet forum (MEF)[10] is developing technical specifications and implementation agreements to promote interoperability and deployment of Carrier Ethernet services across carrier domains. The Metro Ethernet Network (MEN) layer network model specified in the architecture framework defines three layer network components: the Ethernet Services Layer supporting basic Layer 2 Ethernet data communication services; a set of one or more supporting Transport Services Layer(s), and an optional Application Services Layer supporting applications carried on the basic L2 Ethernet services. The layer network model is based on a client/server relationship. In addition, each of these layer networks may be further decomposed into their data, control and management plane components.

Now in order to support the functionalities of Carrier Ethernet service across domains, MEF has defined several technical specifications. Foremost, the Ethernet Virtual Connection (EVC) definition supports the association of UNI reference points for the purpose of delivering an Ethernet flow between subscriber sites across the MEN. There may be one or more subscriber flows mapped to a particular EVC (e.g., there may be more subscriber flows identified by the flow classification rules at the ingress point to a network than EVCs). Furthermore, both point-to-point and multi-point (e.g., mesh, tree) EVC services can be configured. Meanwhile the network interworking network-to-network interface (NI-NNI) is an open interface that supports the extension of transport facilities used to support Ethernet services (and associated EVCs) over an external transport network(s). The NI-NNI is intended to preserve the characteristic information of a subscriber’s flow and also provides a reference point for demarcation between the two MEN service provider interfaces attached via public transport networks, e.g., such as OTN, SDH/SONET, ATM, frame relay, etc. Finally, the service interworking network-to-network interface (SI-NNI) is an interface that supports the interworking of an MEF service with services provided via other service enabling technologies (e.g., Frame Relay, ATM, IP, etc.). The SI-NNI provides a reference point for demarcation between a MEN and another public service network. Examples of other public services networks include ATM, frame relay and IP.

Appendix 2 References

-
- 1 Generalized Multi-Protocol Label Switching (GMPLS) Architecture, RFC 3945, October 2004
 - 2 Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005
 - 3 M. Leech, Key Management Considerations for the TCP MD5 Signature Option, RFC 3562, July 2003
 - 4 J. Moy, OSPF Version 2, RFC 2328, April 1998
 - 5 D. Harrington, R. Presuhn, B. Wijnen, An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks, RFC 3411, December 2002
 - 6 K. Shimoto, et al., "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)" RFC 5212, July 2008
 - 7 G.8080/Y.1304, Architecture for the automatically switched optical network (ASON)
 - 8 Optical Internetworking Forum (OIF), <http://www.oiforum.com/>
 - 9 E. Oki, Tomonori Takeda, J-L Le Roux, A. Farrel. Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering, draft-ietf-pce-inter-layer-frwk-10.txt
 - 10 Metro Ethernet Forum (MEF), <http://metroethernetforum.org/>

Appendix B

Hybrid Multi-Layer Network (Hybrid-MLN) Control Plane Architecture

Hybrid Multi-Layer Network (Hybrid-MLN) Control Plane Architecture Overview

1 Introduction

This document presents an overview of the *Hybrid Multi-Layer Network* (Hybrid-MLN) Control Plane Architecture, developed as part of the Hybrid-MLN project [1]. The key objective here is to build a comprehensive framework to provision a heterogeneous mix of networking data-plane technologies, with a particular focus on future renditions of the DOE *Energy Sciences Net* (ESnet). This initial document focuses on a high-level overview of the Hybrid-MLN architecture.

The primary network configurations addressed in this effort consist of extended network infrastructures with multiple “physical” technology layers or regions, e.g., including IP, Ethernet, SONET/SDH, and optical *wavelength division multiplexing* (WDM). Generally, such setups have been termed as *multi-layer networks* (MLN) and/or *multi-region networks* (MRN) (to be formally defined in Section 2). The goal of the proposed architecture is to define a unified framework that can provision advanced user *service definitions* in such diverse network settings, including connectivity between two end-points (dynamic circuits) as well as connectivity between multiple end-points, i.e., user *application-specific topologies* (AST) or *virtual network topologies* (VNT). In addition, this technology should enable the provision of multi-layer *virtual private network* (VPN) and sophisticated protection and restoration schemes.

The topic of multi-layer/multi-region networking has garnered much attention as of late and many standards bodies are already working on key related aspects. Specifically, some notable players include the *Internet Engineering Task Force* (IETF) *Common Control and Measurement Plane* (CCAMP) and *Path Computation Element* (PCE) working groups, the *Optical Internetworking Forum* (OIF), and the *Telecommunication Standardization Sector* (ITU-T) *Automatic Switched Transport Network* (ASTN). In particular, the recent IETF RFC 5212, titled “*Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)*”, provides a good baseline reference here. Overall, the related architectural constructs and standards developed by these organizations are quite large and diverse. Furthermore, liaison efforts are also underway to synchronize these standardization activities and converge to common interoperable solutions. However, much of this standards work is in its early stages and detailed functional specifications are lacking. Finally, it is noted that some early renditions of hybrid-MLN control have also been implemented in ESnet’s *Science Data Network* (SDN), *Internet2 Dynamic Circuit Network* (DCN), and various other research networks. However, these solutions only provide a subset of the necessary features and capabilities for provisioning hybrid-MLN networks.

In light of the above, the effort herein strikes a balance between leveraging current and future standards efforts (and associated vendor implementations) and developing only the “value-added” features not available now or in the foreseeable future. As such, the key objectives relating to the ongoing Hybrid-MLN project [1] are as follows:

- Leverage the work accomplished and underway in these standards bodies to build a generic framework
- Tailor the architectural components to those actually required by operational *research & education* (R&E) networks, in particular DOE ESnet.
- Extend the functionality in areas where the standards are incomplete

Overall, this report is organized as follows. Section 2 provides a basic overview of the key concepts and terminologies in hybrid-MLN settings. Section 3 then outlines the key architectural components of the proposed framework solution. Subsequently, Section 4 reviews some of the current implementations and identifies which features require additional research and development effort. Finally, Section 5 presents a summary of the existing work in the standards community.

2 Concepts & Terminologies

In order to properly discuss the proposed hybrid-MLN control plane architecture, some related terminologies are introduced first. Specifically, the key concepts here include network regions, network levels, and network element types/adaptation capabilities. Consider the details further:

Network Regions

A network technology type or “region” is uniquely identified by its *interface switching capabilities* (ISC). Specifically, different ISC types are defined according to the data plane switching formats, i.e., layers

- Packet switching capable (PSC) layer, Layer 3: Interfaces handling IP packets
- Layer-2 switching capable (L2SC) layer, Layer 2: Interfaces handling Ethernet frames
- Time-division multiplexing switching capable (TDMSC) layer, Layer 1/1.5: Interfaces handling SONET/SDH timeslots/channels
- Lambda-switching capable (LSC) layer, Layer 1: Interfaces handling optical WDM wavelengths/lambda
- Fiber-switching capable (FSC) layer, Layer 0: Interfaces handling port based switching

Using the above definitions (akin to RFC 5212) a *multi-layer network* (MLN) is defined as a *traffic engineering* (TE) domain comprising of multiple data plane switching layers at either the same ISC or different ISC. Meanwhile, a *multi-region network* (MRN) is special case of a MLN that supports at least two different switching types. Hence a MRN is always an MLN, but an MLN is not necessarily an MRN, e.g., multiple layers can exist in one region.

Network Routing Areas

In extending the above terminology, consider the fact that different network regions will likely belong to different routing areas. Hence a key challenge in control plane architecture design is to be able to support differing granularity of visibility across such boundaries. For example, some MLN’s may provide full link-state exchange/visibility across region boundaries, i.e., peer model construct. Alternatively, others may provide limited “opaque” views, as per more traditional overlay models. Still other variations may achieve a medium between these two possibilities, particularly as controlled by vendor-specific capabilities, e.g., via partial topologies and condensed resource views.

Network Element Types/Adaptation Capabilities

An alternate classification of MLN’s can be done according to the ISC capabilities of their associated network element. Namely, these are identified as follows:

- Single ISC node: Supports a single ISC type.
- Multiple ISC node-simplex type: Supports multiple ISC types, but only one ISC type per physical interface
- Multiple ISC node-hybrid type: Supports multiple ISC types on a single physical interface

Hence a MLN may be constructed from any combination of the above node types. At service instantiation is required across region boundaries, network elements will be required to perform the associated technology adaptations in the data-plane layer(s).

3 Hybrid-MLN Control Plane Architecture Description

In general, a MLN can embody many different combinations of the above-described features, and hence the term “hybrid-MLN”. In order to properly address this diversity, two models are defined for multi-region interaction:

- Common-region model: Two regions interact via a common technology type, e.g., always internetwork at the L2SC layer with mutually agreed *virtual LAN* (VLAN) identifiers.
- Multi-region model: Two regions interact via different technology types. This more complex case requires “cross-layer adaptation” which may be handled by one or both of the participating regions.

4 Hybrid-MLN Specific Architecture

A sample view of the proposed hybrid-MLN architecture is shown in Figure 1 and two key entities are noted here, namely an *inter-domain controller* (IDC) and a *path computation element* (PCE), the latter which is adopted from the IETF (RFC 4655). Owing to its generic design, the architecture makes no further assumptions on intra-region control, and associated solutions can either be based upon MPLS/GMPLS standards or vendor-proprietary offerings (using TL-1 as an example). Now from a functional point of view, three key focus areas are identified in the proposed hybrid-MLN control plane architecture, including 1) topology exchange and routing, 2) path computation, and 3) signaling. These topics are now discussed further in the following sections. Carefully note that additional architectural considerations related to scheduling and *authentication/authorization/accounting* (AAA) functionality are not covered here as the main focus in this overview is related to the multi-layer provisioning aspects. However, the detailed control plane document referenced earlier will indeed cover these topics as well.

Figure 1: Proposed hybrid-MLN architectural overview

4.1 Topology Exchange and Routing

A foremost requirement in hybrid-MLN provisioning is the ability to learn and describe network topologies and resource levels. Indeed, this control plane functionality is deemed vital for implementing subsequent user service provisioning operations. In particular, the description of a MLN topology includes network connectivity, layer-specific TE parameters (resource levels), and any related cross-layer adaptation capabilities. Within a region, various topology exchange mechanisms are being used. For example, in many cases, intra-region MLN routing is done using link-state descriptors, as supported via distributed protocols such as the GMPLS *open shortest path first-TE* (OSPF-TE) [4]. Alternatively, some regions may achieve topological visibility using alternate mechanisms such as centralized routing and/or vendor-proprietary protocols, management systems, or static/pre-configured state.

Now consider the more challenging case of topology exchange/routing in broader inter-region hybrid-MLN settings. Conceptually speaking, the most common exchange models are based upon the peer and overlay modes of operation, with the former providing much better visibility. However, most vendor implementations do not generally support multi-layer peer routing and instead offer more limited proprietary solutions (closer to overlay model). As a result, the proposed hybrid-MLN framework will pursue a more inclusive strategy that leverages all possible mechanisms

to build and maintain real-time inter-region topology/resource views. Namely, associated state information derived from both peer and overlay modes of operation (as well as other possibilities) will be used to construct a unified multi-level intra-domain *TE database* (TEDB). While part of the PCE, the TEDB is also a standalone component, that other systems may interact with directly if needed. The TEDB will be able to flexibly obtain and maintain intra-domain topology information per the combination of techniques described above. In addition, the TEDB will rely on the *inter-domain controller* (IDC) entity, Figure 1, to receive inter-domain topology information. Depending upon the exchange model in place, these messages can represent detailed internal link-state (i.e., peer model), traversing connections (i.e., overlay model), or abstract links via topology abstractions.

Carefully note that routing exchanges are generally aligned with region boundaries, generally limiting state propagation across region boundaries. In these scenarios, the proposed architecture will rely upon the multi-layer PCE (described in Section 3.2) to provide vital “cross-layer” intelligence to control multi-layer networks. This technique of relying on a PCE which has visibility into multiple layers or regions is sometimes referred to as the augmented model or border peer model, where variable amounts of topology data is exchanged outside a single region. In general, these techniques can allow for the required flexibility when dealing with a diverse technology and vendor environments.

4.2 Path Computation

Most existing R&E networks already support path computation capabilities across a single layer, e.g., ESnet OSCARS solution. However, extending path computation to hybrid-MLN settings is much more complicated owing to several key factors including, lack of “global” visibility, “sub-rate” grooming dimensions of multiple network layers, and multiple constraints included as part of computation processes. To address these challenges, the proposed framework will adapt the IETF PCE architecture to MLN settings to achieve end-to-end path optimization. Namely, each domain will have a standalone PCE entity with full topological/resource visibility of all internal layers (in the domain or area under its control). Hence this entity will include the TEDB described in section 3.1, and augment this data with time domain (scheduling), management and policy state to build a very detailed, centralized, domain level view. In addition, the PCE will also have and maintain an inter-domain TEDB with information received from the IDC (Section 3.1).

Using its rich knowledge of domain-internal and domain-external state, the PCE will implement sophisticated algorithms to compute optimal paths across its domain and/or across multiple domains. In turn, this path state will be used to support the IDC (Figure 1) by providing detailed provisioning and signaling inputs across multiple network layers, e.g., explicit route objects. More broadly, the PCE will also provide other key functionalities that will allow the IDC to accomplish its inter-domain routing tasks (Section 3.1), e.g., such as computing abstract topology views, sophisticated path computations, and explicit route objects for MLN signaling and path instantiation. Note that the actual path computation algorithms run by the PCE will be based upon highly-advanced variants of graph-theoretic shortest-path and *k*-shortest path schemes, and other novel schemes, i.e., “on-line” computation.

Overall, PCE design in MLN settings requires much additional research and development. For example, most current PCE implementations have a limited set of multi-layer capabilities and primarily rely upon static information to accomplish very basic path computation. Hence proposed hybrid-MLN PCE architecture must achieve the following: 1) dynamically absorb topology data from multiple sources, 2) combine the TE data with time domain (scheduling), management and policy information, 3) apply TEDB pruning, conversion, and advanced path computation algorithms, 4) generate detailed multi-layer explicit route sequences, and 5) support abstract and user-specific tailored views of network resources based on all available constraints, including the time domain. These are all topics envisioned for future research and development efforts, and many are being addressed in the ongoing Hybrid-MLN project [1].

4.3 Path Control and Signaling

Path control refers to the process in which a PCE dictates a path instantiation. Meanwhile, signaling is the process of physical instantiating the path on all the routers and switches along the end-to-end route. Generally, these two processes are often considered as the same, although the latter is more closely related to switching node control operations. Now within the context of path setup/control herein, path computation results presented for signaling execution will likely require path selection across multiple layers. Hence the proposed hybrid-MLN control architecture will support two path control strategies, “mono-layer” and “multi-layer”. In the former scheme, only “upper-layer” hops will be presented in the path object, i.e., these hops corresponds to lower-layer paths which have

to be expanded at the switch node that crosses from upper layer into the lower layer. This is typical of path control in overlay networks in which connection segments traversing lower layers (such as DWDM) appear as “virtual links” or *forwarding adjacencies* (FA) in higher layer topologies (such as IP, Ethernet). Meanwhile in the latter scheme, both upper- and lower-layer hops will be explicitly presented in the path object.

In addition, the corresponding signaling designs for hybrid-MLN control will also interpret path objects (received from PCE) in order to perform proper switch setup operations at different technology layers. Namely, the layer-specific sub-objects in the path object will be translated into layer-specific operations. Here, carefully note that even though the format of a multi-layer path object can be fixed in the signaling protocol, the translation of the path object into cross-layer adaptation has to take into account the native control methods/commands supported by vendor equipment. This is particularly important in inter-domain environments. For example, when two domains inter-connect as an common-region model, the cross-layer adaptation operations will be hidden by the intra-domain signaling implementations, i.e., signaling protocol messages can either traverse the domain boundary contiguously at a specific layer or stitch the two domains with a common technology by using separate signaling sessions/instantiations. Furthermore when two domains inter-connect as an multi-region model, additional adaptation information will have to be presented by the path object to one side of the peering domains so that an explicit cross-layer adaptation can be executed at its border router. Hence two separate signaling sessions have to be used here in order to “stitch” the two domains into a path, with the adaptation performed on one side of the domain border.

In general, four different types of path signaling interfaces can be identified by summarizing the various deployed network and vendor regions in place today. Specifically, these include 1) GMPLS signaling, 2) MPLS signaling, 3) management plane control, and 4) vendor proprietary control plane signaling. Here, the GMPLS and MPLS regions typically prefer a *user network interface* (UNI) style interaction as opposed to a true peer-to-peer mode of operation for signaling, owing to scalability, operator privacy concerns, and lack of policy enforcement mechanisms.

5 Current Implementation Status and Overview

As noted in Section 1, various aspects of the proposed hybrid-MLN control plane architecture have been implemented in ESnet SDN, Internet2 DCN, etc. However, for the most part, these early implementations only support a limited set of components and capabilities. For example from an intra-domain perspective, each network operates as a single technology region, e.g., ESnet SDN is a router-based MPLS (PSC) region whereas Internet2 DCN is a SONET-based (TDM) region. Hence there is currently no support for further multi-level or multi-region inter-networking. Meanwhile, from an inter-domain perspective, both of these networks perform technology adaptation at their domain boundaries. For example, ESnet SDN adapts PSC interfaces to L2SC (Ethernet) interfaces, and Internet2 DCN adapts TDM interfaces to L2SC (Ethernet) interfaces. Although these adaptations do achieve multi-domain operation, they are unable to provision multiple services based upon different technology characteristics. In other words, the only inter-domain service possible is an L2SC (Ethernet) service. Note however, that ESnet can support traditional MPLS services on an intra-domain basis. In light of the above, the future plan is to develop more complete multi-layer (multi-region, multi-level) functionalities, as proposed in this document.

6 Standardization Overview

Various standardization efforts are looking at the broader MLN/MRN area. Most notably, the IETF standardization body has recently outlined some high-level requirements in RFC 5212, titled “*Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks*.” In general, this ongoing effort is very consistent with the GMPLS framework, and is being developed under the watch of CCAMP working group. For example, several IETF drafts have appeared on MLN/MRN TE link state topology exchange, path control schemes, and signaling triggering mechanisms. In addition, the IETF’s PCE architecture is also closely related to MLN/MRN provisioning.

Meanwhile, the OIF is also working on multi-domain/multi-layer architectures and protocols, with a faster track vision for rapid commercialization. Foremost, the OIF has adopted a hierarchical link state routing model for inter-domain operation, i.e., as part of its *external network-to-network interface* (E-NNI) specification. The OIF framework also introduces the concept of a *routing controller* (RC), Figure 2, which represents one or more routers or switches in a domain. In turn, multiple RC’s can be represented by a higher level RC, which can provide a

summarized view of the lower level topology and hence participate in larger-scale or inter-domain routing. As such, the functionality of an RC overlaps with that of a PCE and thus can be integrated into the IETF PCE architecture.

Finally, some key liaison efforts have also been established to improve architectural compatibility between the IETF standards and those being developed by the ITU-T as part of the *automatically switched transport network* (ASTN) framework. Namely, in ASTN the ITU-T defines the reference points for cross-layer adaptation in the MLN networks. Hence a common focus in both the IETF and ITU-T is to formalize network information models for individual technology layers. Standard provisions for PBB-TE Ethernet and *wavelength switched optical network* (WSN) are currently making good progress, which should help with the implementation of standard MLN/MRN architectures.

As a sample illustration, Figure 2 depicts a “standards-based” view of both MLN and MRN settings operating various domains and layers. Specifically, the use of a border peer model is shown with both standalone and integrated border RC’s. These RC’s have a view of the whole multi-layer topology and can perform cross-layer, cross-domain routing and path control in MPLS-GMPLS and other types of networks. The diagram also shows flexible control plane configurations with varying relationships between router/switch, RC and PCE.

Figure 2: MLN/MRN Standards Based Architecture Illustration

References

- [1] DOE Hybrid Multi-Layer Networks project (Hybrid-MLN, website <http://hybrid.east.isi.edu>)
- [2] IETF Common Control and Measurement Plane (CCAMP) Working Group, www.ietf.org/html.charters/ccamp-charter.html
- [3] K. Shimoto, et al., “Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)” RFC 5212, July 2008

- [4] D. Katz, D. Yeung and K. Kompella, “Traffic Engineering (TE) Extensions to OSPF Version 2” RFC3630, September 2003

Appendix C

Hybrid-MLN DataPlane Test Report and Plans

Hybrid Multilayer Network Data Planes

Network-Layer Testing & Analysis Report

v3.0

**The Hybrid Multi-Layer Network Control for
Emerging Cyber-Infrastructure Project**

<http://hybrid.east.isi.edu>

Table of Contents

1.	Introduction.....	1
2.	Performance Analysis Metrics.....	2
3.	Empirical Test Configurations.....	2
4.	Simulation Test Suite.....	4
4.1.	Network Element Models.....	5
4.2.	End-System Models	6
5.	Single Layer Measurements & Analysis.....	6
5.1.	HOPI-DRAGON Tests.....	6
5.2.	HOPI-DRAGON Cross-Traffic Tests	9
6.	Multi-Layer Measurements & Analysis.....	11
6.1.	UltraScience Net (USN) Tests.....	12
6.2.	UltraScience Net-Energy Sciences Net (USN-ESNet) Tests.....	16
6.3.	HOPI-Abilene-UltraScience Net Tests.....	20
6.4.	HOPI-Abilene-UltraScience Net-Energy Sciences Net Tests	27
7.	Extended Simulations	28
8.	Conclusions.....	31
9.	References.....	32
APPENDIX A.	Nomenclatures & Descriptions.....	33
APPENDIX B.	Empirical Tests Repository.....	35
Appendix B.1	USN and USN-ESNet Tests Archive.....	35
Appendix B.2	HOPI-Sourced Tests Archive.....	35

1. Introduction

Recent years have seen many advances across all networking technology layers. For example, packet-switching regimes (i.e., Layers 2, 3) have seen the adoption of advanced *quality-of-service* (QoS) capabilities, e.g., via frameworks such as *Differentiated Services* (DiffServ) and *multi-protocol label switching* (MPLS). Meanwhile underlying fiber-transport domains have seen much improvement with new standards for *next-generation SONET/SDH* and *dense wavelength division multiplexing* (DWDM). Overall, these advances have yielded unprecedented capacity scalability and management flexibility.

The eScience community has been an early adopter of many of the above-mentioned networking technologies in order to meet its growing application-layer needs [1]. Many next-generation applications are already generating datasets on the order of terabytes-petabytes and these have to be archived and moved across wide-area distances, i.e., owing to the increasingly globalized nature of scientific collaborations. Furthermore, many other application types require stable bandwidths for instrumentation and steering capabilities. It is here that best-effort IP routing networks have fallen short, owing to their inability to provide service guarantees for end-users. As a result, *dedicated bandwidth services*, as supported by the above-detailed technologies (MPLS, Ethernet, SONET, DWDM), offer much promise here for achieving requisite scalability needs.

Currently the eScience community is using a wide range of production and experimental networks. Some of the main infrastructures here include the DOE *UltraScience Network* (USN), DOE *Energy Sciences Network* (ESNet), DOE *Science Data Network* (SDN), Internet2 *Abilene*, Internet2 *Dynamic Circuit Services(DCS)/Hybrid Optical Packet Infrastructure* (HOPI), and NSF *Dynamic Resource Allocation via GMPLS Optical Networks* (DRAGON). In accordance with MPLS/GMPLS nomenclatures, the underlying data plane technologies in these networks can be appropriately classified as one or more of either *packet-switch capable* (PSC), *Layer-2 switch capable* (L2SC), *time-division-multiplex capable* (TDM), or *lambda-switch capable* (LSC), as shown in Table 1. Interested readers are referred to [1] and related references for more details on these network infrastructures.

		Data Plane Technology			
		PSC	L2SC	TDM	LSC
Network	USN		X	X	
	SDN	X	X		X
	ESNet	X			
	I2 Abilene	X			
	I2 DCS/HOPI		X	X	X
	DRAGON	X	X		X

Table 1: eScience network data plane technologies

As the above networks have matured, the broader “hybrid networks” area has emerged as a key focus. Namely, given the increasing scope/scale of scientific collaborations, it is becoming increasingly likely that related applications will have to run across multiple data-plane technologies (Layers 1-2-3) [1]. As a result, the above-listed networks represent a *hybrid multi-layer* environment embodied by the coexistence of many different data-plane technologies, e.g., IP/MPLS (Layer-3), Ethernet VLAN (Layer-2), SONET TDM (Layer 1.5), and lambda wavelength (Layer 1). Therefore end-to-end data paths must now be provisioned across multiple layers and/or concatenations thereof. It is therefore crucial to study related end-to-end performance of such hybrid connections across these diverse infrastructures in order to characterize “end-to-end” service behaviors. Of particular interest is the impact of newer circuit-switching infrastructures—particularly USN—on provisioning dedicated bandwidth services. Nevertheless to date little/no studies have evaluated or quantified the performance of hybrid combinations of such data plane technologies.

Along these lines a detailed comprehensive test plan to address these challenges in real-world eScience network domains was presented in [2]. In particular this effort focuses on analyzing data-plane circuit behaviors across both homogeneous as well as mixed heterogeneous technology domains, i.e., throughput, latency, jitter, loss, etc. Furthermore two key approaches are used here, namely empirical real-world measurement testing as well as detailed software simulation analysis, with the latter being used to corroborate and extend the findings of the former. The overall purpose of this effort, as motivated in [1], is to gain a better understanding of the performance characteristics of long-distance high-capacity paths in a hybrid multi-layer networking environments and compare different networking technologies and concatenations thereof.

This report focuses on data plane performance analysis for hybrid eScience networks and is a follow-up writing to the earlier data-plane overview [1] and test plan [2] documents. Note that this study does not address the techniques utilized to provision hybrid data-plane paths, i.e., control plane design, which is the focus of a future research. Furthermore, the focus herein is strictly on *network-layer* performance only, and the further issue of application layer performance (end-host behaviors) is addressed in a separate companion study, see [3]. The remainder of this document is organized as follows. First Section 2 summarizes the performance analysis metrics used in the study. Subsequently Section 3 gives an overview of the empirical measurement tests whereas Section 4 focuses on the developed simulation suite. Section 6 then presents wide range of empirical measurement results for both single and multi-layer test scenarios. Meanwhile, Section 7 extends upon a broader simulation study of various hypothetical configurations. Finally, Section 8 presents detailed conclusions and recommendations from this overall multi-layer data-plane effort. In addition, various Appendices are also included.

2. Performance Analysis Metrics

To properly quantify and analyze data plane performance, a variety of metrics are used in both the empirical measurements and simulation analysis. A complete list and description is as follows:

- **Average data-rate (throughput):** The average connection throughput measured in bits/second measured at the receiver over the connection lifetime and/or test duration.
- **Average latency (histogram):** The average end-to-end delay of packets arriving at the receiver as measured over the connection lifetime and/or test duration.
- **Jitter profile (histogram):** The average inter-packet delay between packets arriving at the receiver as measured over the connection lifetime and/or test duration.
- **Total packet loss:** The number of sender-injected packets that do not arrive at the receiver as measured over the connection lifetime and/or test duration.

3. Empirical Test Configurations

The tested network configurations comprise a diversified set of data paths within and across a full range of scientific research networks, including USN, ESNet, SDN, Internet2 Abilene, Internet2 HOPI, and DRAGON. The generic detailed multi-layer test configuration is illustrated in Figure 1a (from [2]) and the associated long-distance links are shown more clearly in Figure 2 along with their estimated fiber mileages. These links are used to construct intra-network and inter-network circuits for testing. As per Figure 1a, all test and tested equipment is located at four key locations, namely Sunnyvale, CA (SUNY), Chicago, IL (CHIN), Oak Ridge, TN (ORNL), and Washington D.C. (WASH). In addition, the Seattle, WA, site is also used for various USN and ESNet runs. These sites provide access to all the networks under test and also provide cross-connection/handoff capabilities for inter-network testing purposes.

Overall, Figure 1a shows that a full range of network elements are used in the testing phase to stress multi-layer performance. In particular, these include Layer 3 PSC nodes (i.e., Juniper T640 and Cisco 6509 routers), Layer 2 L2SC nodes (i.e., Force 10 E300/E600 and Raptor switches), Layer 1.5 SONET TDM nodes (i.e., Ciena Coredirector CDCI), and Layer 1 DWDM nodes (i.e., ADVA add-drop multiplexers,

Figure 2: Network links available used for empirical testing phase

Finally, all network-layer measurements are done using specialized Spirent AX/4000 Broadband test system, Figure 1b, end test systems. Although end-user Unix hosts (running various benchmarking applications) can also be used for testing purposes, this approach is not chosen as the main effort herein is to study network-level behaviors, i.e., without host-layer effects. Instead, complete results for such application-layer performance are more appropriately presented in another separate study [3]. The Spirent AX/4000 system provides comprehensive capabilities for broadband traffic generation, full-rate analysis, bit error rate testing, network impairment emulation, and broadband WAN emulation. This system is designed for testing broadband equipment, switches, and networks for proper operation and QoS performance. Specifically, the Spirent test gear provides a very consistent and standardized method to measure circuit performance. Along these lines, the Spirent AX4000 was configured with one 10 Gbps with OC-192POS/BERT/10GBASE-W Ethernet generator/analyzer interface and one 1 Gigabit Ethernet generator/analyzer interface. This allowed a large degree in flexibility in terms of injecting and measuring both the analyzed flow and the various cross traffic data. In particular, the tests distributed four Spirent AX4000 test units through out the network topology, see Figure 1a. Overall, the general procedure for circuit-based measurement with this test gear was as follows [2]:

- Set up the end-to-end circuit path
- Verify basic end-to-end circuit connectivity
- Run data tests across end-to-end circuit (test sequence include multiple flows at varying data rates, MTU sizes, and cross traffic profiles)
- Collect circuit performance metrics including loss rate, jitter, and latency for each test run

For a complete listing of the all test nomenclatures, test-cases, and test-case procedures, interested readers are referred to Appendix A as well as the latest test-plan document [2]. In addition, all detailed archives of the conducted empirical data tests can be found at the main project results website [4], see Appendix B for related access details.

4. Simulation Test Suite

As mentioned earlier, software simulation is used to corroborate the findings of the empirical measurements and also analyze larger hypothetical scenarios. In particular, it is desirable to gauge more

extensive configurations which may be otherwise difficult to recreate in controlled real-world tests. This simulation effort leverages the *OPNET Modeler™* discrete event tool to develop detailed models for the various real-world networks under test, e.g., USN, ESNet, SDN, Abilene, DRAGON, etc. The developed suite leverages the hierarchical *OPNET* modeling framework comprising of sub-networks, networks, nodes, and control processes, Figures 3, 4. Now even though *OPNET Modeler™* provides many existing network and node models, these offerings lack specific QoS features and cannot model the desired intricacies of the real-world networks. As a result detailed network models are coded from the ground up in C/C++ to model all relevant network elements and capabilities (Ethernet switches, MPLS routers, SONET/SDH and DWDM cross-connects, etc) as well as end systems (test systems, work-stations, test applications).

Figure 3: Sample hybrid data plane USN-ESNet simulation scenario in *OPNET Modeler™*

Figure 4: Network element types: I/O packet-switching switching (left), circuit-switching node (right)

4.1. Network Element Models

Detailed network element models are developed to model packet-switching IP/MPLS and Ethernet devices as well as circuit-switching SONET/SDH and DWDM nodes. Specifically, the packet-switching nodes have been designed around a generic *input/output* (I/O) buffering switch node model with full QoS-support. This core model implements processes for input/output link buffer management and scheduling along with switching fabric functions (see Figure 4) and can be tailored for either Layer 2 or Layer operation by wrapping around more specific “packet-layer” encapsulations and switching functionalities, e.g., IP packets

and routing lookup tables, MPLS shim headers and label switching tables, Ethernet input/output port tables, etc. Furthermore, users can directly specify any associated node-level parameters such as buffer sizes, scheduling policies, packet buffering/discard policies, switching delays, input-output table associations, etc.

Meanwhile the circuit-switching node models are designed to implement the key data-plane capabilities of SONET/SDH (TDM) and optical DWDM (LSC) switching nodes. As these elements do not perform any “data-packet” inspection per say, the key implemented functionalities include edge mappings and time-slot bypass and add/drop switching. In particular the latter models the *generic framing procedure* (GFP) scheme in which 1 Gigabit Ethernet and 10 Gigabit Ethernet interfaces are mapped to their closest STS-1 level equivalents, e.g., 21 and 171 STS-1, respectively. Note that these procedures can add some latency as packets are buffered to await the next available STS-1 timeslot.

4.2. End-System Models

A variety of end-system models are developed to model the various testing tools used in the empirical tests (Section 3). These testing tools either comprise of end-system hosts (running specific benchmarking software such as file transfer, ICMP ping, TCPMON, UDPMON, etc) or specialized packet-generation gears, e.g., Spirent AX/4000 [2]. However, given the general difficulty of implementing every possible testing tool in the simulation suite, particularly end-host benchmark software, only those most-accurately measuring network-layer performance coded in the simulation suite. These include models for the Spirent AX/4000 box and work-station applications such as one-way file transfers, roundtrip ICMP ping, and roundtrip TCPMON/UDPMON. Again end-users have full flexibility in specifying associated end-system test parameters such as maximum segment sizes, timer values, port speeds, end-system latencies, etc. Full provisions are also included for capturing numerous end-to-end statistics (e.g., throughput, delay, jitter, loss, etc) for optional post-processing in *MATLAB*.

5. Single Layer Measurements & Analysis

Initially, the first set of measurements are done for single layer networks (Ethernet only) using the Spirent AX/4000 test systems. These tests insert a reference Ethernet packet test stream via sourcing test system and then measure associated performance metrics at the receiving test system, e.g., end-to-end delay, inter-arrival times, packet bit-rate/throughputs, and packet loss (Section 2). Here all generated reference packet streams have fixed inter-packet spacings (i.e., fixed transmission rate) and traverse across specific path routes and vendor systems. The main goal here is to assess the performance of the end-to-end stream for varying traffic conditions, e.g., packet transmission rates/sizes, with/without interfering cross-traffic, etc.

5.1. HOPI-DRAGON Tests

Initially, basic loopback testing is done at the Ethernet switching (L2SC) layer. Specifically, runs are done using the DRAGON network in the Washington D.C. area by looping back a connection between two Raptor switches located at the Arlington and McLean sites, i.e., tests 12a, 12c, 12d in [2]. Specifically, the data-plane path is as follows (Figure 5):

Spirent source → Arlington-Raptor → McLean-Raptor (loopback) → Arlington-Raptor → Spirent receiver

Figure 5: HOPI loop-back test scenario (test 12a, 12c, 12d)

and the peak packet insertion rates (at the Gigabit Ethernet interface on the Spirent tester) are varied between 100, 500, and 800 Mbps. Figures 6 and 7 plot the measured inter-arrival and end-to-end packet delay histograms for two extreme MTU sizes at 100 Mbps transmission rates. Specifically these include small 64 byte packets—reflective of remote steering applications—and larger 8,000 byte packets—reflective of large-scale petabyte/exabyte file transfer applications. Note that the corresponding idealized inter-packet transmission times here are 8.3 μ s and 642.4 μ s, respectively. Overall, the measured inter-arrival delay histograms (Figures 5a, 6a) show very tight distributions about their ideal means. Meanwhile the end-to-end delay distributions are also very tight, but the mean increases notably for the larger 8,000 byte MTU value owing to increased packet transmission time. No losses are observed here and the respective packet goodputs are measured at 75.99 Mbps (64 bytes) and 99.67 Mbps (8,000 bytes). This is inline with expectations as larger MTU sizes yield lower relative framing overheads for datagrams.

Figure 6: DRAGON loop-back (64 byte MTU, 100 Mbps sending rate):
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Figure 7: DRAGON loop-back (8,000 byte MTU, 100 Mbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

To corroborate the above findings with software simulation, the HOPI loopback test-case scenario is tested using discrete event simulation. Here, the simulated inter-arrival mean values are plotted in Figure 7 along with their commensurate measured empirical values. As expected, the inter-arrival time scales linearly with the MTU size and very strong correlation is observed between the analytical and real-world results. In addition, the simulated inter-packet time distribution is plotted for 8,000 byte MTU size in Figure 8 and closely matches the measured unimodal distribution in Figure 7a.

Figure 8: DRAGON loop-back packet inter-arrival times (simulation vs. empirical)

Figure 9: DRAGON loop-back simulated packet inter-arrival histogram (8,000 byte MTU, 100 Mbps)

Subsequent tests extend upon the above scenario by introducing *multi-vendor* diversity at the Ethernet switching layer. Namely, a uni-directional test is done in which a reference Ethernet packet stream is transmitted across both the DRAGON and HOPI networks using two different Ethernet switches, i.e., Raptor and Force10 E600 (as shown in Figure 10). Again the Spirent test systems are connected using Gigabit Ethernet ports whereas the switches are connected using 10 Gigabit Ethernet ports, maintaining the bottleneck link rate at 1.0 Gbps (i.e., tests 13b, 13c in [2]):

Spirent source → Arlington-Raptor → McLean-Raptor (loopback) →
Washington-Force10 → Spirent receiver

Figure 10: DRAGON-HOPI multi-vendor test scenario (test 13b, 13c)

The resulting inter-arrival and end-to-end packet delay histograms for smaller 64 byte and larger 8,000 byte MTU sizes are plotted for 100 Mbps packet transmission rates in Figures 11 and 12, respectively. Here a notable difference is observed versus the single vendor tests (Figures 6 and 7). Namely, there is a slight increase in the variability of packet inter-arrival times (i.e., increased jitter), particularly for the smaller packet size. Moreover both cases no longer exhibit the unimodal inter-packet delay distributions observed in the single vendor runs. The resulting impact on end-to-end delay histograms is a net increase in the maximum-minimum spread to about 30 μ s, i.e., Figures 11b, 12b. The likely reason for this increased variability is vendor-specific port interfacing issues. However the measured packet goodputs are not

impacted here and average about 77.32 Mbps (64 bytes MTU) and 99.613 (8,000 bytes MTU). Note that these results are difficult to model via simulation without exact knowledge of vendor port-level interfacing implementations. As a result related simulation studies are not done here.

Figure 11: DRAGON-HOPI multi-vendor test (64 byte MTU, 100 Mbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Figure 12: DRAGON-HOPI multi-vendor test (8,000 byte MTU, 100 Mbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

5.2. HOPI-DRAGON Cross-Traffic Tests

Single-layer HOPI-based scenarios are also tested in conjunction with interfering Ethernet traffic streams. Namely, a competing stream is generated using the Spirent AX/4000 tester and outputted on a higher-speed 10 Gigabit Ethernet interface. Here both the reference and interfering packet streams are mapped to their own individual *virtual LAN* (VLAN) on to the same output Ethernet port using common buffering. Namely, no Ethernet-layer QoS mechanisms are used at the switch and only rate enforcing/policing is done at the edge via the Spirent devices. Furthermore the larger speed interface is used to generate cross-traffic in order to capture the effects of more diverse traffic conditions. Initial cross-traffic runs are conducted for the HOPI loop-back scenario in Figure 5 with light cross-traffic loads (test 12a-xt). For example the inter-arrival and end-to-end packet delay histograms for a reference 100 Mbps stream (64 byte MTU) in the presence of 800 Mbps interfering cross-traffic with large 8,000 byte MTU sizes are shown in Figure 13. Corresponding simulation results are also presented in Figure 14 for this scenario. Overall these empirical and simulation findings show a clear change in packet delay behaviors, with the inter-arrival distribution exhibiting a wider spread. The key reason for this is the fact that smaller 64 byte packets (from the reference test stream) can potentially be buffered behind larger 8,000 byte packets (from the interfering stream) and then released in quick succession at full 10 Gbps rates. This behavior is termed as “*packet compression*” since the output packet stream exhibits compressed inter-arrival times, e.g., as evidenced the

large histogram spike at about 720 ns inter-arrival times in Figure 13a. This value corresponds to 64 byte transmission at 10 Gbps (after accounting for Ethernet packet encoding overheads).

Figure 13: DRAGON loop-back (64 byte MTU, 100 Mbps) w. 800 Mbps cross-traffic (8,000 byte MTU)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Figure 14: DRAGON loop-back (64 byte MTU, 100 Mbps) w. 800 Mbps cross-traffic (8,000 byte MTU)
Simulated (a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Furthermore, the measured variation packet inter-arrival and end-to-end delays (i.e., maximum-minimum spreads) are also plotted in Figure 15 for various cross-traffic packet sizes at 800 Mbps speeds. In general these findings show that end-to-end delay variations remain within the tens of microseconds range for link capacity loadings below 90% and this is generally acceptable for most eScience applications. In all of these tests the reference stream also gets near-optimal throughput, i.e., 100 Mbps minus relative overheads depending upon chosen MTU size. Note that more extensive testing of cross-traffic scenarios is also conducted for various multi-layer scenarios (Section 6.2) and using detailed simulation analysis (Section 7).

Figure 15: DRAGON loop-back delay ranges (max-min) w. 800 Mbps cross-traffic (8,000 byte MTU)

Various measurement runs are now done using heavier cross-traffic loads. For example the DRAGON-HOPI multi-vendor setup in Figure 10 is tested using a 800 Mbps reference stream along with 8 Gbps interfering cross-traffic stream (8,000 byte MTU), i.e., test 13c-xt. This setup gives slightly over 80% link loading and sample results for large 7,700 byte MTU reference packets are plotted in Figure 16. The end-to-end delay histogram here shows a similar spread as that seen with lighter cross-traffic loads and 8,000 byte MTU sizes (Figure 13b), i.e., under $30 \mu\text{s}$ range. Meanwhile, the packet inter-arrival histogram shows a large concentration about the theoretical value of about $80 \mu\text{s}$. Nevertheless, carefully note that no packet compression effects are seen here for the larger reference packet size. Additional runs with smaller 64 byte MTU reference packet sizes (not show), however, do show packet compression.

Figure 16: DRAGON-HOPI test (7,700 MTU, 800 Mbps) w. 8 Gbps cross-traffic (8,000 byte MTU)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

6. Multi-Layer Measurements & Analysis

A wide range of multi-layer scenarios tested next in order to gauge the interactions of multiple link-layer technologies. Expectedly, many of these scenarios span larger geographic domains and are more reflective of the real-world transfers in eScience application settings. Moreover these scenarios are all necessarily multi-vendor in nature. Furthermore, owing to the inherent packet-to-circuit mappings across many layer boundaries, these scenarios are expected to yield increased variations in packet delay and jitter performances. Initial measurement tests strictly focus on the advanced DOE USN infrastructure and then subsequent runs investigate joint USN-ESNet data-plane interfacing. Finally, the last set of measurement tests are conducted over a mix of the USN, ESNet, HOPI, and Abilene infrastructures. In all of the setups

tested, Gigabit Ethernet interfaces are mapped to full 21 STS-1 equivalents whereas the 10 Gigabit Ethernet interfaces are mapped to 171 STS-1 equivalents, i.e., as detailed in test plan document [2].

6.1. UltraScience Net (USN) Tests

Initial multi-layer USN tests focus on *Ethernet over SONET* (EoS) performance across Ethernet and SONET switches, i.e., L2SC and TDMSC nodes. Here basic loopback measurements are done by mapping the Spirent-generated test stream over a long-distance Chicago-Sunnyvale link, as per Figure 17. Note that the actual loopback here is performed at the SONET tributary level and comprises of the following path

Spirent source → Chicago-Force 10 → Chicago-CDCI → Sunnyvale-CDCI
Sunnyvale-CDCI → Chicago-CDCI → Chicago-Force 10 → Spirent receiver

Figure 17: Chicago-Sunnyvale direct loopback test scenario

The above route is tested for several packet sizes at full 1.0 Gbps transmissions rates and the measured histograms (inter-arrival, end-to-end) are shown for two extreme MTU sizes (64 bytes and 8,000 bytes) in Figures 18 and 19. In general these results show that both packet sizes tend to experience similar levels of delay jitter, i.e., in the tens of microseconds range (30-50 μ s). These values arise due to port interface buffering between Force 10 Ethernet switches and CDCI SONET devices, i.e., Ethernet packet mapping onto SONET tributaries. However, by and large, the inter-arrival times are generally well-centered about their ideal respective means. For example most 64 byte packets experience about 0.5 μ s spacing whereas larger 8,000 byte packets see approximately 80 μ s (i.e., after accounting for 8/10 byte Ethernet encoding).

Figure 18: Chicago-Sunnyvale direct TDM loopback (64 byte MTU, 1 Gbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Figure 19: Chicago-Sunnyvale direct TDM loopback (8000 byte MTU, 1 Gbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Next the above USN scenario is modified slightly by changing the sourcing location to ORNL and also using an Ethernet switch at the remote (Sunnyvale) location to implement loopback at the Ethernet level. This revised data-plane path is shown in Figure 20 and comprises of the following network elements:

Spirent source → Chicago-Force 10 → Chicago-CDCI → Sunnyvale-CDCI →
Sunnyvale-Force 10 → Chicago-CDCI → Chicago-Force 10 → Spirent receiver

Figure 20: Direct ORNL-Sunnyvale loopback test scenario w. Ethernet switch loopback

The above scenario is tested for various packet sizes, including 64, 512, 1,500, and 8,000 bytes. For all of these runs no packet losses are observed even when outputting at full 1.0 Gbps line rate at the Spirent tester. Moreover some sample results for 64 and 8,000 byte MTU values are presented in Figures 21 and 22. In comparing these histograms with those in Figures 18 and 19 (i.e., shorter Chicago-Sunnyvale loopback), some key similarities are observed in the distribution type and ranges. Foremost, the end-to-end delay variation for 64 byte packets is still largely contained within 50 μ s of the mean (i.e., compare Figures 18b, 21b). Similarly end-to-end delay variation for 8,000 byte packets is also contained within about 40 μ s of the mean (i.e., compare Figures 19b, 22b). However, as expected, there is a noticeable increase in end-to-end delay, approximately 12 ms, owing to the increased distance of this loopback test, i.e., sourcing node changed to ORNL from Chicago.

Figure 21: Direct ORNL-Sunnyvale loopback test w. Ethernet switch (64 byte MTU, 1 Gbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Figure 22: Direct ORNL-Sunnyvale loopback test w. Ethernet switch (8000 byte MTU, 1 Gbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Next a USN loop-back measurement is conducted for a much longer ORNL-Sunnyvale route using intermediate SONET CDCI switches at Chicago and Seattle, as shown in Figure 23. The aim here is to gauge Ethernet performance over extended TDM link concatenations (and intermediate TDM-layer switching), up to six in this particular case. Hence the full end-to-end node sequence here is given as:

Spirent source → ORNL-Force 10 → ORNL-CDCI → Chicago-CDCI → Seattle-CDCI →
Sunnyvale-CDCI → Seattle-CDCI → Chicago-CDCI → ORNL-CDCI →
ORNL-Force 10 → Spirent receiver

Figure 23: ORNL-Chicago-Seattle-Sunnyvale loopback test scenario

Again various packet sizes are tested and some sample histogram measurements for 64 and 8,000 byte MTU packets are shown in Figures 24 and 25. These histograms are compared with those in Figures 18 and 19 (i.e., Chicago-Sunnyvale loopback) and 21 and 22 (ORNL-Sunnyvale loopback). Foremost, no packet losses are observed in all the runs, and as expected, these results show general agreement with the above in terms of distribution spreads. However, in comparing Figure 25a with Figure 22a, there is a slight reduction in the variability of inter-arrival times for larger 8,000 MTU packets, i.e., fewer “spikes”. Furthermore, the end-to-end delay variation for larger MTU size is also notably smaller than those in the loopback tests discussed above, i.e., 10 μ s range, on the order of multi-vendor Ethernet-only tests (Figure 12). This result shows that the concatenation of multiple TDM framed links in the end-to-end path can help control and limit delay variability.

Figure 24: ORNL-Chicago-Seattle-Sunnyvale loopback test (64 byte MTU, 1 Gbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Figure 25: ORNL-Chicago-Seattle-Sunnyvale loopback test (8000 byte MTU, 1 Gbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Finally, additional loopback tests are done from ORNL to Sunnyvale using Chicago as an intermediate site, as shown in Figure 26. However there are some key differences here versus the earlier ORNL-Sunnyvale test configuration in Figure 23. Foremost this scenario uses fewer SONET links, and akin to that in Figure 20, implements per-site loopback via Ethernet switches, i.e., loopback is done at the Ethernet packet and not SONET tributary level. Hence the full end-to-end route is given as:

Spirent source → ORNL-Force 10 → ORNL-CDCI → Chicago-CDCI → Chicago-Force 10 →
Chicago-CDCI → Sunnyvale-CDCI → Sunnyvale-Force 10 →
Sunnyvale-CDCI → ORNL-CDCI → ORNL-Force 10 → Spirent receiver

Figure 26: ORNL-Seattle-Sunnyvale loopback test scenario w. Ethernet switch loopback

For this configuration a whole range of packet sizes are tested, ranging from 64 to 8,000 bytes in steps of 1,000 bytes. For the smallest 64 byte MTU size, the overall delay histograms (not shown) are largely identical to those in Figures 18, 21, and 24 (with the only exception being different end-to-end delays).

Meanwhile, for the larger 1,000-byte range MTU sizes the associated delay distributions also exhibit much the same “spiky” patterns as those seen in Figures 19, 22, and 25, e.g., a sample result for 8,000 byte packets is shown in Figure 27. Perhaps the only key noticeable difference here is a slightly larger spread in the end-to-end packet delays, i.e., increasing from a few tens of microseconds to about the 60-80 μ s range. This is expected due to the additional Ethernet switches in the path, i.e., at Chicago and Sunnyvale. Overall, these results confirm that line-rate EoS interfacing works well and the addition of multiple TDM segments on an end-to-end path adds very little/no discernable delay packet stream distortion.

Figure 27: ORNL-Chicago-Seattle-Sunnyvale loopback test (8,000 byte MTU, 1 Gbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

6.2. UltraScience Net-Energy Sciences Net (USN-ESNet) Tests

The next set of tests investigate more complex USN and ESNet data-plane combinations, essentially traversing across three complete networking layers, i.e., Layers 1, 2, and 3. Here the first scenario modifies the basic USN Chicago-Sunnyvale loopback test by routing the return path over an ESNet tunnel using IP/MPLS routers, as depicted in Figure 28. In particular, this setup contrasts with the test in Figure 17 which uses USN SONET links for both forward and reverse paths. Namely, the end-to-end node sequence here is now given as:

Spirent source → Chicago-Force 10 → Chicago-CDCI → Sunnyvale-CDCI → Sunnyvale-Force 10 →
Sunnyvale-T640 → Chicago-T640 → Chicago-Force 10 → Spirent receiver

Figure 28: Chicago-Sunnyvale loopback (USN-ESNet) test scenario

Some sample results for this scenario are presented in Figures 29 and 30. For example, inter-arrival and end-to-end distributions are presented in Figure 29 for small 64 byte packets and show general agreement with those for the SONET-only USN runs. Perhaps the only slight change is the slightly larger increase in inter-arrival time spread, i.e., more arriving packets with 10 μ s spacing (comparing Figures 18a and 29a). In addition, no packet losses were observed even at full gigabit transmission rates with this smaller MTU

size. Meanwhile Figure 30 shows sample delay histograms for larger 1,000 and 1,500 byte MTU packets, respectively. In comparing these results versus those with the USN-only runs (and larger 8,000 byte packets, i.e., Figures 19b, 22b, and 27b) a notable increase is seen in the end-to-end delay, with variability exceeding the 0.1 ms (100 μ s) range. In addition packet losses are also observed (approximately 0.25%) and there is a notable increase in end-to-end delays over the 64 byte MTU runs, e.g., mean value of about 67.97 ms in Figure 29b versus over 84 ms in Figures 30a and 30b.

Figure 29: Chicago-Sunnyvale loopback w. MPLS tunnel (64 byte MTU, 1 Gbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Figure 30: Chicago-Sunnyvale loopback w. MPLS tunnel, end-to-end delay histograms for 1 Gbps sending
(a) 1,000 bytes MTU, (b) 1,500 bytes MTU

To further investigate USN-ESNet performance, the next scenario modifies the ORNL-Seattle-Sunnyvale loopback test in Figure 26 by adding an IP/MPLS tunnel between Chicago and Sunnyvale. This entails additional mappings between Ethernet Force 10 switches, as shown in Figure 31, and the resultant end-to-end node sequence is as follows:

Spirent source → ORNL-Force 10 → ORNL-CDCI → Chicago-CDCI → Chicago-Force 10 →
Chicago-T640 → Sunnyvale-T640 → Sunnyvale-Force 10 → Sunnyvale-CDCI →
ORNL-CDCI → ORNL-Force 10 → Spirent receiver

Figure 31: ORNL-Seattle-Sunnyvale loopback test scenario w. ESNet tunnel

A variety of packet sizes and transmission rates are tested for this configuration as well. Foremost, results with the smaller 64 byte MTU size, shown in Figure 32, indicate much the same pattern as earlier runs in USN and USN-ESNet. Meanwhile, end-to-end delay histograms for larger 1,500 byte packets are shown in Figure 33 for two different capacity loadings. Namely, the findings for 97-98% loading (i.e., 970-980 Mbps transmission rates at the Spirent tester, Figure 33a) show good performance, with no observed packet losses and mean end-to-end delays in the same range as those for 64 byte packets. However with slightly faster 99-100% link capacity loading, much more deleterious impacts are seen. Foremost there are non-negligible packet losses again (averaging about 0.25%, akin to results with Figure 28) along with a sizeable non-linear increase in end-to-end packet delays, i.e., by almost 17 ms. These loss behaviors, in particular, will be of key concern to higher-layer protocols and applications, particularly TCP-based which modulate transmission rates accordingly. More importantly these finding contrast with those for multiple SONET link concatenations (e.g., ORNL-Chicago-Seattle-Sunnyvale loopback in Figure 23) which yielded much tighter delay variations and no losses, e.g., compare Figures 25b and 33b for same round-trip propagation distance. It is further possible that the concatenation of multiple IP/MPLS (ESNet) tunnels may cause such variabilities to grow.

Figure 32: ORNL-Seattle-Sunnyvale loopback w. MPLS tunnel (64 byte MTU, 1 Gbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Figure 33: ORNL-Seattle-Sunnyvale loopback w. MPLS tunnel
End-to-end delay histograms for 1,500 byte MTU (a) 97% link loading, (b) 100% link loading

The final tests consider a very long distance double loopback scenario using path segments on USN and ESNet, as shown in Figure 34. Namely this configuration is designed to loop back twice from Chicago-Sunnyvale along both TDM and MPLS tunnels. The full node sequence is listed as follows:

Spirent source → Chicago-Force 10 → Chicago-T640 → Sunnyvale-T640 → Sunnyvale-Force 10 →
Sunnyvale-CDCI → Chicago-CDCI → Sunnyvale-CDCI → Sunnyvale-Force 10 →
Sunnyvale-T640 → Chicago-T640 → Chicago-Force 10 → Spirent receiver

Figure 34: Chicago-Sunnyvale double loopback scenario

The above scenario is tested for various packet sizes and transmission rates. First, basic runs are done at full line rate using 64 byte MTU and the findings plotted in Figure 35. As expected, the end-to-end delays are much larger here, about 136 ms. Additionally, the distribution in Figure 35b shows a slightly extended tail (as compared to those in Figures 29b, 32b), owing to the increased buffering capacity along this path, i.e., more IP/MPLS hops. Nevertheless, no packet losses are observed here. Next, commensurate histogram results are shown for larger 8,000 byte packets transmitted at full 1.0 Gbps line rate, Figure 36. Akin to the results in Figure 30b and 33b, there is a significant non-linear increase in end-to-end delay, i.e., by about 30 ms, almost double the 17 ms increase observed in the previous ORNL-Seattle-Sunnyvale USN-ESNet loopback test (Figure 31). In addition packet losses are also observed. Overall, these findings show that it is problematic to run MPLS tunnel segments at over 98% of their peak capacity, particularly for MTU sizes over 512 bytes, owing to Ethernet-to-MPLS framing overheads. This will require appropriate edge policing and possibly core QoS enforcement mechanisms, e.g., per-class buffering, scheduling, etc. Inevitably, the latter per-hop mechanisms will entail higher system costs, the absence of which can result in serious performance degradations. These issues are further investigated in the simulation study (Section 7).

Figure 35: Chicago-Sunnyvale double loopback (64 byte MTU, 1 Gbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Figure 36: Chicago-Sunnyvale double loopback (8,000 byte MTU, 1 Gbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

6.3. HOPI-Abilene-UltraScience Net Tests

Extended multi-layer test measurements are also done by combining paths across the USN, HOPI, and Abilene networks. Namely Ethernet reference packet streams are now originated from the Washington D.C. area and measured at different locations. In the first such test, a reference stream is sent to Chicago via a series of diverse network elements (test 11b-2 in [2], see also Figure 37):

Spirent source → Washington-Force 10 → Washington-Juniper T640 → Chicago-Juniper T640 →
Chicago-Glimmer Glass → Chicago-Force 10 → Spirent receiver

From the above, the end-to-end bottleneck link rate here is determined as 1.0 Gbps (Spirent tester port). Again several packet sizes and transmission rates are tested. For example, initial runs are done for 100 Mbps transmission rates and 64 byte MTU values, and sample packet inter-arrival and end-to-end delay distribution histograms are plotted in Figure 38. In addition larger 8,000 byte MTU values are also tested at faster 800 Mbps transmission rates, Figure 39. In comparing the above results versus those for single “Ethernet-only” switching (Figures 6 and 7), some key differences are noted in the histogram distributions. Specifically, smaller 64 byte MTU sizes show some signs of packet stream compression, as evidenced by the large spike close to 0 μ s in Figure 38a. Furthermore there is a slight increase in inter-packet delay variation (on the order of tens of microseconds) for the faster 800 Mbps/8,000 byte MTU tests as well. These effects most likely arise due to Ethernet port buffering issues between Layer 2 (Force 10 switches) and Layer 3 (Juniper T640 routers) devices. Nevertheless, the overall end-to-end delays remain tightly

distributed about a fixed mean (approximately equivalent to the end-to-end delay) with relatively small variation, i.e., under $50 \mu\text{s}$. This performance is sufficient for most eScience applications.

Figure 37: Multi-layer HOPI-Abilene-USN test scenario (test 11b)

Figure 38: Multi-layer HOPI-Abilene-USN test (64 byte MTU, 100 Mbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Figure 39: Multi-layer HOPI-Abilene-USN test (8,000 byte MTU, 800 Mbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

To further stress the test configuration of Figure 37, cross-traffic is also inserted along the path to gauge the impact on a reference test stream. Namely, 3 Gbps worth of interfering traffic (8,000 bytes MTU) is now inserted along the full route, i.e., via Washington D.C. Force 10 switch all the way to the Chicago Force 10

switch. The resulting inter-arrival and end-to-end delay histograms are shown in Figures 40 and 41 for the same 64 and 8,000 byte MTU values. In comparing these results with those for the non-interfering scenarios (i.e., Figures 38 and 39), is seen that packet inter-arrival times are somewhat more dispersed, as evidenced by more distribution spread over 40 μ s. This yields a slight increase in the mean end-to-end delay as well, approximately about 6 μ s (i.e., compare Figure 38b with 40b). Nevertheless these increases are relatively minor and will not impact most overlying eScience applications. Overall, this is expected as the scenario is not a heavy-load scenario, running under 50% link capacity loading.

Figure 40: HOPI-Abilene-USN test (64 byte MTU, 100 Mbps) w. 3 Gbps cross-traffic (8,000 byte MTU)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Figure 41: HOPI-Abilene-USN test (8,000 byte MTU, 800 Mbps) w. 3 Gbps cross-traffic (8,000 byte MTU)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Carefully note that the path in Figure 37 simply terminates on USN equipment (i.e., Chicago Force 10 switch) and does not traverse any USN links per say. Hence a subsequent configuration is designed and tested in which traffic is further routed over the 700 mile SONET USN link from Chicago to Oak Ridge (i.e., tests 18c in [2]). This scenario is shown in Figure 42 and has the following node sequence:

Spirent source → Washington-Force 10 → Washington-Juniper T640 → Chicago-Juniper T640 →
Chicago-Force 10 → Chicago-Glimmer Glass → Chicago-Force 10 →
Chicago-CDCI → Oak Ridge-CDCI → Oak Ridge-Force 10 → Spirent receiver

Figure 42: Multi-layer HOPI-Abilene-USN test scenario (test 18c)

The above path embodies a true multi-layer scenario as it traverses Layer 1 (DWDM), Layer 1.5 (SONET), Layer 2 (Ethernet), and Layer 3 (IP/MPLS) devices. Again, the resulting inter-arrival and end-to-end delay histograms for this run are plotted in Figures 43 and 44 for two diverse MTU sizes (64 bytes, 8,000 bytes at 100 Mbps sending rate). Overall the runs show much similarity with the earlier results in Figures 38 and 39 (above) in terms of uni-modal end-to-end delay distributions with very tight standard deviations, i.e., low inter-packet arrival jitter. However slightly higher inter-packet delay variation is seen with the smaller 64 byte MTU sizes as compared to the shorter USN-HOPI run in Figure 38 (i.e., less relative packet compression). Overall, these findings re-confirm that Ethernet streams can readily be mapped over longer-distance L2SC SONET/SDH networks with tolerable impacts on end-to-end delay/jitter performance (akin to results in Section 6.2). Note that many simulations runs are also done to verify the measured performance of these multi-layer *non-cross-traffic* scenarios. However, detailed results are not presented here as the scenarios are relatively straightforward and the associated results show very close matches with the measured values.

Figure 43: Multi-layer HOPI-Abilene-USN test (64 byte MTU, 100 Mbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Figure 44: HOPI-Abilene-USN test (8,000 byte MTU, 100 Mbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Next, additional cross-traffic test runs are done for the above HOPI-Abilene-ESNet-USN configuration of Figure 42 (Section 6.1), i.e., test 18c-xt [2]. Again an interfering packet stream is inserted at the source (i.e., Washington D.C. Force 10 switch, Figure 42) and routed across all end-to-end path links. This stream has a rate of 3 Gbps and uses 8,000 byte MTU packets. Some sample delay histogram distributions from this scenario are shown in Figures 45 and 46 and contrasted with those for the unloaded settings (Figures 43 and 44). Here it is seen that the addition of light cross-traffic results again gives a slight increase in the respective delay performances, on the order of tens of microseconds, e.g., peak inter-arrival times increase from about $60 \mu\text{s}$ to slightly over $100 \mu\text{s}$ for 64 byte MTU. Although packet compression effects distort the inter-packet spacing for smaller 64 byte MTU sizes, the mean value for the larger 8,000 byte packet sizes is still centered about the ideal $640 \mu\text{s}$ value. Furthermore, no packet losses are observed in any of the runs and the test stream achieves close to full throughput, i.e., 100 Mbps minus associated framing overheads. Carefully note that even though the above data-plane path traverses many nodes, inter-stream packet contention really only occurs at the first node, i.e., Washington D.C. Force 10 switch.

Figure 45: HOPI-Abilene-USN tests (64 byte MTU, 100 Mbps) w. 3 Gbps cross-traffic (8,000 byte MTU)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Figure 46: HOPI-Abilene-USN tests (8000 byte MTU, 100 Mbps) w. 3 Gbps cross-traffic (8000 byte MTU)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

To further study this cross-traffic scenario, simulation runs are also done to corroborate against measured values. For example, simulated inter-arrival delay histograms are plotted for in Figure 47 for two MTU values (64 bytes, 8,000 bytes). In general these results show a good match versus the measured values, e.g., packet compression effects captured in Figure 45a for smaller 64 byte packets and max-min inter-arrival range of 620-665 μ s in Figure 45b. Note that the measured values do, however, show some slightly larger outlier values owing to the relatively shorter simulation intervals chosen in the simulation runs, i.e., 1-2 seconds at gigabit line rates only.

Figure 47: HOPI-Abilene-USN simulated packet inter-arrival histograms (100 Mbps sending rate)
(a) 64 byte MTU, (b) 8,000 byte MTU

As mentioned earlier, inter-stream contention only occurs at the (common) outbound link at the first Ethernet node since all of the cross-traffic scenarios use a single interfering traffic stream. In other words, subsequent downstream nodes/links will not see any contention between the reference and cross-traffic streams as their respective packet orderings are already determined at the first link, i.e., traffic smoothing. Hence these tests may not fully capture broader more realistic settings in which bandwidth contention is likely to occur at many nodes (i.e., L2SC, PSC) along an end-to-end path. These scenarios are now considered in more detail in the subsequent simulation study (Section 7).

Finally, inter-layer performance is tested more thoroughly for coast-to-coast distances and higher transmission speeds. In one example a data-plane path is routed from HOPI-Abilene out across long-

distance USN SONET links from Chicago to Sunnyvale, as shown in Figure 48. The key modification in this test (versus Figure 42) is the inclusion of full 10 Gbps interfaces and longer TDM segments. Specifically, increasing transmission rates to the multi-gigabit level is more reflective of high-end file transfer scenarios. The commensurate full data path here is (i.e., test 19a in [2]):

Spirent source → Washington-Force 10 → Chicago-Force 10 → Chicago-Glimmer Glass →
Chicago-Force 10 → Chicago-CDCI → Seattle-CDCI → Sunnyvale-CDCI →
Seattle-CDCI → Chicago-CDCI → ORNL-CDCI → ORNL-Force 10 →
Spirent receiver (GigE)

Figure 48: Coast-to-coast HOPI-Abilene-USN test (test 19a)

Figure 49: HOPI-USN-Abilene-ESNet test (8,000 byte MTU, 9.1 Gbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

The resulting packet delay histograms for 9.1 Gbps sending rate are shown in Figure 49 for large 8,000 byte packet sizes, i.e., almost two orders increase in sending rate versus 100 Mbps. These measurements indicate that packet delay distributions are well maintained at higher packet rates, e.g., end-to-end delay distribution largely contained within 40 μ s (akin to USN and USN-ESNet runs in Sections 6.1, 6.2). However, from Figure 49a there is a small increase in the number of inter-packet delay arrivals exceeding 20 μ s. However, the net effect on the end-to-end delay histogram is not that significant, with the max-min range staying within the 30 μ s range (similar to Figures 19b and 22b). In addition the packet goodput rate

is measured at 9,077,395.44 Mbps and no packet losses are observed. In general, this test run again shows that the addition of multiple TDM segments (TDMSC) along a hybrid data-plane path does not yield noticeable packet stream delay variations.

6.4. HOPI-Abilene-UltraScience Net-Energy Sciences Net Tests

The final set of multi-layer tests are done over a very long concatenated route spanning across network nodes/links in HOPI, Abilene, USN, and ESNet. This path route is shown in Figure 50 and originates in Washington D.C. and loops back via Sunnyvale (i.e., test 20b-2 in [2]). A key distinction of this test is the use of diverse directional paths on the Chicago-Sunnyvale portion, i.e., USN in forward direction, ESNet tunnel in the reverse direction. In particular, the full data path is as follows:

Spirent source → Washington-Force 10 → Washington-Juniper T640 → Chicago-Juniper T640 → Chicago-Force 10 → Chicago-Glimmer Glass → Chicago-Force 10 → Chicago-Cisco 6509 → Seattle-Juniper T640 → Sunnyvale-Juniper T640 → Sunnyvale-Force 10 → Sunnyvale-CDCI → Seattle-CDCI → Chicago-CDCI → Chicago-Force 10 → Chicago-Juniper T640 → Washington-Juniper T640 → Washington-Force 10 → Spirent receiver

Figure 50: Coast-to-coast Washington D.C. to Sunnyvale roundtrip test scenario (test 20b)

In all, the above end-to-end data plane path has a length of nearly 7,000 route miles and its bottleneck capacity is limited to 1.0 Gbps, i.e., due to the Gigabit Ethernet interface between the Juniper T640 and Force 10 E300 switches at the Sunnyvale location. This scenario is stress-tested for high transmission rates and the sample delay histograms for 995 Mbps insertion rates and 1,500 byte MTU sizes are shown in Figure 51. Again the findings show that overall packet delay variations are contained to the 10 μ s range. Furthermore the associated packet goodput rate is measured at 983,696.33 Mbps and no packet losses are observed. Nevertheless, additional runs with slightly higher 998 Mbps insertion rates show some slight packet loss, averaging about 0.06% or one packet loss per roughly 1,800 packets. This is expected as 998

Mbps is close to the maximum theoretical achievable rate on an MPLS-framed Gigabit Ethernet channel, i.e., see IETF RFC 4448 for Layer 2 transport over MPLS networks [5]. Moreover, similar losses were observed in many of the USN-ESNet tests involving IP/MPLS tunnel segments (see Section 6.2)

Figure 51: Coast-to-coast roundtrip test (8,000 byte MTU, 100 Mbps sending rate)
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

7. Extended Simulations

To better gauge realistic multi-layer networking environments, some extended simulation studies conducted. In particular the key aim here is to test pertinent configurations which may be otherwise difficult to measure in controlled empirical settings for a variety of reasons, e.g., such as lack of multiple distinct traffic generation systems, limited control over resources and traffic in production networks, setup complexity across multiple national sites/locations, etc. Initially, various extended simulation scenarios are developed by modifying some of the existing real-world scenarios of Section 6. Subsequently alternate hypothetical test scenarios are also designed using a “clean-slate” approach.

Figure 52: Modified HOPI-Abilene-USN network with two cross-traffic streams (100 Mbps sending)

Simulations are first conducted for the HOPI-Abilene-USN cross-traffic scenario of Figure 20 to gauge overall delay performance for a wider range of interfering traffic rates and packet MTU sizes. Specifically, interfering packet inter-arrival times are now randomly distributed using an exponential distribution, with the means appropriately chosen to vary the average sending rate ranging between 1.0 and 9.8 Gbps. More importantly, to increase packet-level contention and stress delay behaviors, *two* equal-rate Ethernet cross-traffic VLAN streams are inserted at two different locations, i.e., Washington D. C. and Chicago Force 10

switches in the HOPI network, as denoted by the dotted red lines in Figure 52. These cross-traffic streams are assigned their own individual VLAN identifiers but are commonly buffered and allowed to share the same outbound Ethernet port, i.e., no Ethernet-layer QoS mechanisms invoked.

The resulting spread (i.e., maximum-minimum range) in inter-packet arrival times and end-to-end delays observed for the fixed 100 Mbps reference stream are plotted in Figure 53 for various reference MTU sizes and interfering cross-traffic sending rates (8,000 byte cross-traffic MTU). As expected, there is a sharp increase in inter-packet and end-to-end delay as the aggregate link loading approaches the bottleneck 10 Gbps output link rate at the Ethernet switches. These behaviors arise from the increased buffering being done at the Layer 2/3 devices, i.e., Ethernet switches, IP/MPLS routers. In addition it is seen that larger cross-traffic MTU sizes have much more deleterious impacts, driving overall end-to-end delay variations to the 0.5 ms range—with just two contention sites.

Figure 53: Modified HOPI-Abilene-USN network (100 Mbps sending rate)
(a) 512 byte cross-traffic MTU, (b) 8,000 byte cross-traffic MTU

To further observe these delay behaviors, sample inter-arrival and end-to-end packet delay histograms are plotted in Figure 54 for 8,000 byte MTU packets and 9.5 Gbps/8,000 byte MTU cross traffic loading (i.e., 96% aggregate link loading). These results show a very notable increase in delays with the addition of contention at just one more hop, e.g., compare (simulated) distribution spreads between single and double link contention inter-arrival times in Figures 47b and 54a. More importantly the maximum end-to-end delay rises to about 7.27 ms, more than 0.5 ms over the fixed propagation delay of 6.74 ms. Given that interfering traffic is exponentially-distributed here it is likely that the addition of more (realistic) bursty on-off sources and or “heavy-tail” long-range dependent patterns will yield even higher delay distortions.

Figure 54: Modified HOPI-Abilene-USN network
Reference 100 Mbps sending rate/8,000 byte MTU, w. 9.5 Gbps cross-traffic
(a) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Figure 55: Coast-to-coast multi-node Ethernet test network

Next a more hypothetical coast-to-coast Ethernet IP/MPLS data-plane path is considered, as shown in Figure 55. This route runs from New York City (NYC) to Sunnyvale and comprises of 7 nodes and 6 links as shown. All switching devices have 10 Gbps link rates and implement *early packet discard* (EPD) buffering (100 packets per link buffering). Meanwhile, the reference source sends data at close to full gigabit line rate, i.e., 995 Mbps, using 8,000 byte MTU packets, yielding a theoretical inter-packet spacing of about $64.41 \mu\text{s}$. Furthermore, per-hop contention is inserted at all transit nodes/links in order to stress performance. The simulated delay histograms (for inter-arrival and end-to-end) for this scenario are shown in Figure 56 for relatively high cross-traffic loads of 8.5 Gbps (i.e., total link loading approaching 95%). These results show end-to-end delays increasing by over 1 ms, albeit no packet losses are observed. Clearly, practical settings can see such values rise to even higher levels, thereby proving problematic for higher-end services.

Figure 56: Multi-hop simulation with no Ethernet-layer QoS support
Reference 995 Mbps sending rate, 8,000 byte MTU, w. 8.8 Gbps cross-traffic
(b) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

To resolve the challenges posed by heavy/large MTU cross-traffic, subsequent simulations are done using per-hop QoS support mechanisms for the reference source packet traffic. Namely, all source packets originating at NYC are tagged to the highest priority class at the network edge and then granted the required 995 Mbps capacity at each L2SC path node via a *weighted fair-queuing* (WFQ) scheduler. The resulting inter-arrival and end-to-end delay distributions are now plotted in Figure 57 and show some very

drastic improvements. Namely, there is a very sharp “tightening” of the histogram behaviors, i.e., inter-arrival time distribution spread drops to tens of microseconds and the maximum end-to-end delay falls to about 12.24 ms (i.e., only a few tens of microseconds above fixed propagation delay). Simulations are also done by replacing 50% of the hops in Figure 55 with SONET segments. Although the findings are not shown here, the results show even tighter delay variations. These results confirm that the use of state-of-the-art QoS techniques at the L2SC and/or PSC layers can effectively emulate “TDM-like” performances. However these mechanisms are generally very costly to deploy at multi-gigabit line rates, particularly 10 Gbps interface cards. Instead it is likely more cost-effective to deploy next-generation “channelized” SONET systems—such as the CDCI devices used in USN—which are capable of highly-stringent delay/jitter support and flexibly matching Ethernet traffic demands in 1.5 or 50 Mbps increments.

Figure 57: Multi-hop simulation with Ethernet-layer QoS support
Reference 995 Mbps sending rate, 8,000 byte MTU, w. 8.8 Gbps cross-traffic
(c) Packet inter-arrival time histogram, (b) end-to-end packet delay histogram

Overall, the above simulations have explored some relevant and hypothetical scenarios in multi-layer data-plane settings, and the findings largely corroborate the observations of the extensions empirical measurements-based test phase. This effort opens up the scope for further more detailed analyses, in particular the study of highly bursty traffic behaviors, e.g., long-range dependant, on multi-segment data-plane delay/loss performance.

8. Conclusions

This report presents detailed network-layer performance evaluation results for hybrid data-plane paths in eScience cyber-infrastructures. Specifically the effort has focused on testing paths spanning multiple networks, technologies, and tunnel concatenations. The related studies are done using both empirical measurements on real-world networks as well as software-based simulation of hypothetical scenarios. Overall, some of the major conclusions from this effort include the following:

- 1) In general, all of the tested networking technologies (PSC, L2SC, TDM, LSC) performed well both individually and when concatenated together, provided that input traffic is not driven close to or beyond the bottleneck path capacity. Therefore it is possible to build dedicated stable connection-oriented network services across using any one and/or a combination of these technologies. In general application requirements and cost will be the main factors in determining which data-plane technology is the best choice.
- 2) Some key differences were observed between the various networking layer technologies. Namely, MPLS tunnels (over ESNet) exhibited notably different delay behaviors versus dedicated circuit-paths (over USN) when bandwidth usage was driven at or close to maximum levels, i.e., end-to-end bottleneck capacity. Specifically, there were non-negligible packet losses along with non-

- linear increases in end-to-end latencies (tens of microseconds) owing to relatively large interface line card buffers on IP/MPLS routers. More importantly these effects were also observed regularly on composite tunnels paths with one or more IP/MPLS tunnel segments. In general, these deviations can be controlled either by proper management of ingress traffic and/or applying QoS bandwidth enforcement mechanisms at routers/switches along the path.
- 3) The inclusion of multiple TDM-based path segments (USN) added little/no notable increase in end-to-end delay variations. Instead the variations were largely fixed, as per associated Ethernet-over-SONET packet mappings, i.e., about 50 μ s. This implies that provisioning paths over TDM-based infrastructures is most germane for applications requiring extremely stringent guarantees on latency, jitter, and bandwidth protection. Some examples here include remote instrumentation and control, synchronized computer-to-computer transfers, real-time data mirroring, etc.
 - 4) Ingress traffic policing and management are crucial for building guaranteed bandwidth services across hybrid networks. Specifically, proper “edge-based” mechanisms can help maintain high levels of performance and also greatly simplify end-to-end services management, particularly if there are IP/MPLS tunnel segments along an end-to-end route. However, some form of QoS enforcement is generally necessary at all IP/MPLS routers (segments) along an end-to-end hybrid path to guarantee performance.
 - 5) Inter-layer cross-connections can be achieved in a reasonable manner by “stitching” together different network layer technologies. However, since the interface between two networks must meet at a common layer, Ethernet presents the most ubiquitous and natural choice. Specifically, Ethernet VLAN mappings were regularly used to interconnect paths across the USN, ESNet, HOPI, DRAGON, and Abilene networks. Furthermore, associated demarcations can be chosen based upon untagged Ethernet links or tagged Ethernet links. However when using the latter approach, care must be taken to coordinate the available VLAN across all the networks involved.
 - 6) Vendor/equipment differences at the Ethernet switching layer (L2SC) can also introduce a small amount of variation in delay performance. In general, however, these levels are bounded to the lower tens-of-microseconds range and will not be of concern to most eScience applications.
 - 7) A key concern is the potential impact of traffic “burstiness” on end-to-end delay and loss profiles, i.e., both for reference and interfering cross-traffic streams. This issue was not addressed in this effort and will inevitably have implications for associated ingress policing mechanisms. Therefore the study of traffic burstiness can be a future topic for both empirical and simulation efforts.

9. References

- [1] “Hybrid Multilayer Network Data Planes: Test, Analysis, and Modeling Plan,” Version 7.0, December 2006.
- [2] “Hybrid Multilayer Network Data Planes: Test Configuration”, Version 6.0, March 2007.
- [3] “Hybrid Multilayer Network Data Planes: Application-Layer Testing & Analysis,” Version 1.0, in preparation (see also N. Rao, et al, “Measurements on Hybrid Dedicated Bandwidth Connections,” *IEEE INFOCOM 2007 Workshop on High Speed Networks*, Anchorage, Alaska, May 2007.
- [4] Empirical data-plane tests results repository, <http://hpn.east.isi.edu/dataplane/sprint-test-data/>.
- [5] L. Martini, *et al*, “Encapsulation Methods for Transport of Ethernet over MPLS Networks”, *IETF Internet Request for Comments (RFC) 4448*, April 2006.

APPENDIX A. Nomenclatures & Descriptions

As noted in [1], network data planes can be constructed of multiple segments each using different data plane technology. Along these lines, the following MPLS/GMPLS-based nomenclature is used for the purposes of identifying data plane technologies and describing test results:

- Packet-Switch Capable (PSC)
- Layer-2 Switch Capable (L2SC)
- Time-Division-Multiplex Capable (TDM)
- Lambda-Switch Capable (LSC)

Carefully note that a distinction is made between the data plane transport technology and the client port framing technology. As a result, this test plan will specifically focus on the mapping of Ethernet client access ports onto different data plane transport technologies. Given the above background, commensurate nomenclature formats are defined to describe the data plane paths under test. Specifically, this nomenclature specifies the network name, data-plane switching technology, and ingress/egress access framing type as follows:

network [accessframing:dataplane:accessframing]

where the following values are possible for the above parameters:

- *network* - ESNet, sdn, abilene, i2abilene, i2hopi, i3dcs, usn, dragon
- *accessframing* – Ethernet (SONET, Infiniband not included)
- *dataplane* - psc, l2sc, tdm, lsc (where psc is a PSC path with QoS applied to the LSP)

For example, a sample multi-layer, multi-domain, multi-technology circuit path is given as follows:

i2dcs [ethernet:l2sc:tdm:l2sc:ethernet] : sdn [ethernet:psc:ethernet]

A similar nomenclature will be used to indicate the physical path (i.e., route) a circuit traverses. For instance a circuit of type *ESNet [ethernet:psc:ethernet]* whose endpoints are in Washington and Sunnyvale might have a path description given by *ESNet [WASH:NYCM:CHIN:SUNV]*. Namely, the implication here is that the WASH and SUNV locations will contain the edge port where the client Ethernet access ports are located. Meanwhile the NYCM and CHIN nodes are transit nodes which simply switch through the MPLS LSP. In summary the combination of the circuit type and circuit path descriptors will fully define the specific circuit under test. The generic formats for these two descriptors are:

- Circuit type: *network [accessframing:dataplane:accessframing]*
- Circuit path: *network [ingressloc:transitnodes:egressloc]*

As an example, Figure 58 illustrates a sample data-plane test path traversing across I2 DCS, ESNet, and USN. The formal description of this path is as follows:

Circuit type: *usn [ethernet:tdm:ethernet] : i2dcs [ethernet:tdm:ethernet] : ESNet [ethernet:psc:ethernet] : usn [ethernet:tdm:ethernet]*

Circuit path: *usn [ORNL:CHIN] : i2dcs [CHIN:WASH] : esnet [WASH:CHIN] : usn [CHIN:STTL:SUNV]*

This inter-network path has four segments which are concatenated at three cross-layer mapping points. Namely, the mapping between USN and I2 DCS is performed by the Ciena multi-service CDCI platform at CHIN using the Ethernet-over-SONET mapping feature. Meanwhile the mapping between I2 DCS and

ESNet is performed by the Juniper router at WASH and CHIN using the *circuit cross-connect* (CCC) feature. Finally, the mapping between ESNet and USN is performed via sequential back-to-back Juniper router CCC and Ciena Ethernet-over-SONET mappings at CHIN.

Figure 58: Sample inter-layer data-plane path traversing across I2 DCS, ESNet, and USN

The above nomenclature is utilized to describe and identify the circuits tested. The specific network paths, including the detailed list of network elements traversed, are presented in [2].

APPENDIX B. Empirical Tests Repository

All of the empirical results have been carefully indexed and archived for future purposes. These repositories are also available to the public via open web-based URL access. Specifically, there are two key repositories, one for the USN and USN-ESNet tests (i.e., Sections 6.1 and 6.2) and another for the HOPI-sourced tests (Sections 5.1, 5.2, 6.3, and 6.4). The related access details are as follows:

Appendix B.1 USN and USN-ESNet Tests Archive

The measurement data for USN and USN-ESNet tests is available at the following URL website listing:
<http://www.csm.ornl.gov/ultranet/SpirentMeasurements/>.

Appendix B.2 HOPI-Sourced Tests Archive

All measurement data for tests originating on the Washington D. C. area HOPI network are available via the following URL website <http://hpn.east.isi.edu/dataplane/sprint-test-data/>. These runs are all indexed by their respective test numbers, as detailed in the test plan document [2]. Moreover, the results for a specific test can also be found by composing its specific URL as follows:

[http://hpn.east.isi.edu/dataplane/sprint-test-data/test <#>/<description>/index.html](http://hpn.east.isi.edu/dataplane/sprint-test-data/test<#>/<description>/index.html)

where “#” refers to the test number, e.g., 12a, 18c-xt, 19c, etc, and “description” details related MTU packet sizes and sending rates. For example, the results for test 19c with an MTU size of 1,500 bytes and bandwidth of 999 Mbps can be found by composing the following URL:
<http://hpn.east.isi.edu/dataplane/sprint-test-data/test19c/19c-mtu1500-bw999/index.html>.

Hybrid Multilayer Network Data Planes: Test, Analysis, and Modeling Plan

v7.0

**The Hybrid Multi-Layer Network Control for
Emerging Cyber-Infrastructure Project**

<http://hybrid.east.isi.edu>

Table of Contents

1.	Introduction	1
2.	Hybrid Network Data Planes	2
2.1.	Network Data Plane Overview	2
2.2.	Data Plane Description and Identification Nomenclature	4
3.	Test Equipment	5
3.1.	Spirent AX/4000 Test Broadband Test System.....	5
3.2.	End System Testing.....	6
3.3.	Measurement Suite	6
4.	Network Test Plan.....	6
4.1.	Intra-Network Paths	7
4.2.	Inter-Network Paths	8
4.3.	Test Sequence.....	9
5.	Analysis Framework	10
6.	Modeling and Simulation Development Plan	12
6.1.	Network Element Models.....	13
6.2.	End-System Models	13
6.3.	Simulation Test Plan	14
7.	Summary	14
	APPENDIX A: Network and Test Equipment Configuration	15
	APPENDIX B: Test Procedures and Data Collection	16
	APPENDIX C: Segmented Regression Method	17

1. Introduction

Hybrid network architectures are rapidly gaining acceptance within the high performance network community as a promising path forward to meet the increasing demands of large-scale science users and associated networked applications. The hybrid network concept promises the advantages of both the current best-effort Internet Protocol (IP) service and dedicated deterministic end-to-end network services. The details of these "deterministic services" are under active discussion and development at this time; however, fundamentally these are expected to be provisioned as circuits or connection-oriented services. Multiple network technologies are available to provision these connection-oriented network services. In terms of "data-plane" these services can be provisioned at different layers:

- IP router-based Multiprotocol Label Switching (MPLS) tunnels
- Ethernet VLAN based Circuits
- Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH) circuits
- Wavelength Division Multiplexing (WDM) connections

A provisioned connection may consist of segments of different technologies, which are selected by factors including performance, costs, available physical resources (such as fiber plants), current network equipment, vendor relationships, user requirements, and others.

Currently there exists no comprehensive data and analysis techniques to evaluate and quantify the performance of connections provisioned by these technologies or their combinations. To address these challenges, this document describes a detailed data collection, analysis and model development plan to evaluate and characterize the performance of connections composed of different data plane technologies. Our target connections include multiple paths, each of which uses a homogeneous technology, as well as those traversing a mix of heterogeneous technology layers. The overall objectives of these activities are to provide qualitative and quantitative answers to the following questions:

- What are the performance characteristics (in terms of jitter, packet loss, and latency) of circuits provisioned using individual data plane technologies?
- What are the performance characteristics of an end-to-end circuit composed of multiple individual circuits, each spanning multiple network domains and utilizing multiple data plane technologies?
- What is the performance of an end-to-end circuit which is constructed via the hierarchical nesting of multiple individual circuits provisioned via different data plane technologies?
- How do we quantify and characterized the performance of the data plane technologies in a manner such that network designers and end users can make decisions in terms of what is best suited to their needs and objectives? [e.g effect of adding SONET GFP segment to Ethernet]
- Can we build network analytical and simulation models that correlate the empirical findings of real world data collection to a reasonably close degree?
- Can we use these data plane analytical and simulation models to further predict performance and behaviors in more generalized hybrid network scenarios and help with proper design efforts?

Toward this end, data will be collected on the following real-world networks:

- DOE Energy Sciences Network (ESNet)
- DOE Science Data Network (SDN)
- UltraScience Network (USN)
- Internet2 Abilene
- Internet2 Network/Hybrid Optical Packet Infrastructure (HOPI)
- Dynamic Resource Allocation via GMPLS Optical Networks (DRAGON)

The above-listed networks represent a hybrid multi-layer networking environment, and embody the coexistence of many different data-plane technologies, e.g., including IP/MPLS (Layer-3), Ethernet VLAN (Layer-2), SONET TDM (Layer 1.5), and lambda wavelength (Layer 1). Hence the planned data plane tests are intended to investigate the latency, loss and jitter performance of circuits using these different data plane technologies. In addition, performance tests of circuits constructed via combination and concatenation of multiple networks and technologies will also be conducted. The purpose of these tests is to provide real-world information that can help network architects, service providers, and users to understand the performance characteristics of the various data plane technologies and circuit construction options.

In addition to empirical (i.e., real-world) data collection and analysis, a multi-technology, multi-layer data plane simulation model will also be developed. This simulation model will be used to further evaluate (verify) results from data plane test measurements as well as form the basis for hybrid network design, analysis, and optimization. This simulation capability will leverage the *OPNET ModelerTM* simulation tool development environment.

It should be noted that the activities outlined in this document do not address the techniques utilized to provision these data-plane paths. Instead, future activities conducted by this group will address such issues including multi-domain, multi-level, multi-service *control planes* which allow for the dynamic provision of end-to-end hybrid network services. This document solely focuses on data plane performance evaluation.

The remainder of this document is organized as follows. First, Section 2 discusses the various hybrid network data planes under consideration. Subsequently, Section 3 describes the test equipment to be used and then Section 4 discusses the specific network involved and associated data paths to be tested. Finally, Section 5 provides a description of the Modeling and Simulation Plan and Section 6 presents an overall summary. In addition, Appendix A lists some details on the network element configurations and test equipment attachments, whereas Appendix B contains details on the test procedures and data collection.

2. Hybrid Network Data Planes

The "data plane" options to be considered in this test plan are as follows:

- IP router-based Multiprotocol Label Switching (MPLS)
- Ethernet VLAN based Circuits
- Synchronous Optical Network/Synchronous Digital Hierarchy (SONET/SDH)
- Wavelength Division Multiplexing (WDM)

In many cases a specific network data plane will be constructed of multiple segments, most likely with each using different data plane technology. Carefully note that a distinction is made between the data plane transport technology and the client port framing technology. For example, a SONET/SDH switch will often have Ethernet ports for client access and/or IP routers may commonly provide SONET port mappings. Hence the end result is that a client-attached Ethernet frame is mapped on to SONET/SDH TDM slots and transported across the network to its endpoint which will be another Ethernet device. While the client attachment point and perceived transport mechanism is Ethernet frames, the actual data transport is SONET/SDH. The same scenario can be described for WDM and IP routed data plane transport. The key point to stress here is that access port framing is separate and distinct from data plane transport technology. As a result, this test plan will specifically focus on Ethernet client access ports which are mapped to the different data plane transport technologies.

Now in order to be precise for the data plane testing plan, we will follow commonly-used GMPLS nomenclature for the purposes of identifying data plane technologies and describing test results:

- Packet-Switch Capable (PSC)
- Layer-2 Switch Capable (L2SC)
- Time-Division-Multiplex Capable (TDM)
- Lambda-Switch Capable (LSC)

A brief description of the networks to be tested and their respective data plane technologies and capabilities is provided below.

2.1. Network Data Plane Overview

Below is a summary of the various networks to be tested and the associated data plane technologies.

Energy Sciences Network (ESNet)

ESNet consists of IP routers with the capability to set up circuit paths via the use of MPLS LSPs. This is a PSC network with quality of service (QoS) feature support.

ESNet Science Data Network (SDN)

SDN will begin coming on-line in early 2007 and will consist of an Ethernet switching infrastructure. Namely, Ethernet switches can be operated in a router (Layer 3) mode or as Ethernet switches (Layer 2). As a result it will be possible to establish circuit paths via MPLS LSPs (PSC) or Ethernet VLANs (L2SC).

Carefully note that SDN will use an underlying vendor-provided WDM infrastructure. This WDM infrastructure is utilized solely to provide interconnection between Ethernet switches and the current plan does not include the provision of WDM or wavelength services (other than to build the Ethernet layer). As a result, SDN does not operate as a WDM or LSC switched network. However, it will be possible to test the LSC data plane used to interconnect the Ethernet switches, and this will be included in the test plan.

UltraScience Network (USN)

USN consists of two data plane layers, i.e., SONET and Ethernet. The SONET switches allow establishment of circuit paths based the provision of TDM time slots. Meanwhile the overlaid Ethernet switches allow provisioning of path-specific Ethernet VLAN configurations. As a result USN includes the capability to provision circuits based on SONET (TDM) and/or Ethernet VLANs (L2SC). Note that there are also some multi-layer Ethernet-over-SONET (EoS) configurations that can be provisioned as well.

Carefully note that USN will use an underlying vendor-provided WDM infrastructure. This WDM infrastructure is utilized solely to provide interconnection between Ethernet switches and the current plan does not include the provision of WDM or wavelength services (other than to build the Ethernet layer). As a result, USN does not operate as a WDM or LSC switched network. However, it will be possible to test the LSC data plane used to interconnect the Ethernet switches, and this will be included in the test plan.

Internet2 Abilene

Abilene consists of IP routers with the capability to set up circuit paths via the use of MPLS LSPs. This is a PSC network.

The New Internet2 Network

The Internet2 Network will begin coming on line in late 2006 and will consist of three data plane layers, i.e., WDM, SONET, and Ethernet. The WDM data plane will allow the creation of circuits which map Ethernet or SONET client interfaces directly onto wavelength transport. The SONET switches will allow establishment of circuit paths based the provision of TDM time slots. Finally, the Ethernet layer will enable the provisioning of path specific Ethernet VLAN configurations. The Ethernet layer network is also referred to as HOPI (Hybrid Optical Packet Infrastructure). As a result The Internet2 Network will include the capability to provision circuits based on WDM (LSC), SONET (TDM), and Ethernet VLANs (L2SC). There are also some other multi-layer configurations which can be provisioned as well. In addition, the Abilene network will run directly ontop of the WDM layer of the new network.

Dynamic Resource Allocation via GMPLS Optical Networks (DRAGON)

The DRAGON network supports three layers, i.e., optical switching layer, Gigabit Ethernet switching layer, and an IP routing layer. As a result DRAGON can provision circuits based on SONET (TDM), Ethernet VLANs (L2SC), and IP MPLS (PSC). There are also some other multi-layer configurations which will be discussed in detail in the test plan section (Section 4) under the topic of multi-layer circuit construction.

The table below summarizes the data plane technologies available in each of the networks to be tested.

Table 1: Available data plane technologies and networks

		Data Plane Technology			
		PSC	L2SC	TDM	LSC
Network	ESNet	X			
	USN		X	X	
	Abilene	X			
	I2Net		X	X	X
	DRAGON	X	X		X

2.2. Data Plane Description and Identification Nomenclature

Given the above background, commensurate data plane nomenclature formats will be defined to describe the various data plane paths under test. Specifically, this nomenclature will specify the network name, data-plane switching technology, and ingress/egress access framing type, i.e., as follows:

network [accessframing:dataplane:accessframing]

where the following values are possible for the above parameters:

- *network* - esnet, sdn, abilene, I2Net, usn, dragon
- *accessframing* - ethernet, sonet (not included in this test plan), infiniband (not included in this test plan)
- *dataplane* - psc, pscq, l2sc, tdm, lsc (where pscq is a PSC path with QoS applied to the LSP)

As an example, a circuit provisioned across ESNet in which the end-user connects to an Ethernet port on a Juniper router which is mapped into an LSP is therefore identified as:

esnet [ethernet:psc:ethernet]

Furthermore, if this same circuit was "glued" (i.e., stitched) to an L2SC circuit traversing across HOPI (I2Net), the combined end-to-end path spanning two networks would be identified as this:

esnet [ethernet:psc:ethernet] : hopi [ethernet:l2sc:ethernet]

The above is an example of a circuit which spans two networks and two data plane technologies, i.e., multi-domain, and multi-technology. The two individual circuits are "glued" (i.e., stitched) together using a peering edge. If there is a layer transition in between two access framing points, this would be referred to a multi-layer circuit. For instance, here is an example of a multi-layer end-to-end circuit:

i2net [ethernet:l2sc:tdm:l2sc:ethernet]

In addition, the following is an example of a multi-layer, multi-domain, multi-technology circuit path:

i2net [ethernet:l2sc:tdm:l2sc:ethernet] : sdn [ethernet:psc:ethernet]

A similar nomenclature will be used to indicate the physical path a circuit traverses. For instance a circuit of type *esnet [ethernet:psc:ethernet]* whose endpoints are in Washington and Sunnyvale might have a path description given by *esnet [WASH:NYCM:CHIN:SUNV]*. Namely, the implication here is that the WASH and SUNV locations will contain the edge port where the client Ethernet access ports are located. Meanwhile the NYCM and CHIN nodes are transit nodes which simply switch through the MPLS LSP. In summary, the combination of the circuit type and circuit path descriptors will fully define the specific circuit under test. The generic formats for these two descriptors are:

- Circuit type: *network [accessframing:dataplane:accessframing]*
- Circuit path: *network [ingressloc:transitnodes:egressloc]*

The above nomenclature will be used to describe and identify the circuits to be tested. The specific network paths, including the detailed list of network elements traversed, are presented in Section 4

3. Test Equipment

Note: We still need to coordinate with Spirent and see if it is possible for them to support this test program.

After specific circuits are configured two types of tests will be utilized for data collection. The first will be via use of a Spirent AX/4000 Broadband Test System. The second will be via Linux hosts. This test equipment and the associated testing plan is included in the below sections. Note that the planned activities will be coordinated with Spirent in order ensure proper support for the test plan.

3.1. Spirent AX/4000 Test Broadband Test System

The Spirent AX/4000 Broadband Test System, Figure 1, provides complete broadband traffic generation, full-rate analysis, bit error rate testing, network impairment emulation, and broadband WAN emulation. This system is designed for testing broadband equipment, switches, and networks for proper operation and QoS performance. It uses modular components, allowing custom configuration for various test applications. The primary hardware components are the chassis (with a control module), an external controller, test modules, and port interfaces. With the exception of the power switch and front panel status indicators, the AX/4000 equipment is operated completely from the controller using the controller software. The controller software runs on IBM PC compatible computers operating in the Microsoft Windows 2000, ME, or XP graphical environment or a Sun Solaris workstation operating on a UNIX platform.

Figure 1 Spirent AX/4000 Broadband Test System in an XLP Portable Chassis

In the proposed test plan setup, four Spirent AX/4000 XLP portable chassis systems will be used, with each deploying the following configuration.

- One 10 Gbps with OC-192POS/BERT/10GBASE-W Ethernet generator/ analyzer/interface
- Two Gigabit Ethernet 1310nm LX SFP generator/analyzer/interface
- One AX/4000 XLP GPS Ethernet E4 controller

Furthermore, with this system three interface options are available, i.e., UNI-PHY 1550 IP/Ethernet, 1550nm Ethernet WAN and 1310nm Ethernet WAN. Overall, the Spirent test gear will provide a very consistent, standardized method to measure the circuit performance. The general procedure for circuit-based measurement with this test gear will be as follows:

- Set up the end-to-end circuit path
- Verify basic end-to-end circuit connectivity
- Run data tests across end-to-end circuit (test sequence will include multiple flows at varying rates including 1 Gbps, 5 Gbps, and 10 Gbps and also varying MTU sizes)
- Collect circuit performance metrics including loss rate, jitter, and latency for each test run

The specific tests descriptions, data collected, and end-to-end path combinations to be exercised are further detailed in Section 4.

3.2. End System Testing

In addition, each end-to-end circuit will also be tested via a standard-configuration Linux system. Even though the performance data collected via this method will not be as precise/accurate as that measured by the Spirent gear, it will nevertheless still provide some very useful information. Namely, this effort will give a realistic opportunity to evaluate how an operational end-system will see data transfers and how it can leverage (or mask) variations in raw network performance. These tests will be conducted using direct memory to memory transfer performance applications like iperf or nuttcp and will include testing of TCP and UDP protocols. Identically configured end-systems will be used for all tests. The general end system configuration will be as follows:

- Linux Fedora Core 5
- Dual Processor MotherBoard
- At Least 2 GByte system memory
- Syskonnet 1 PCI-66 Gigabit Ethernet Card
- Chelsio 10 PCI-X 133 Gigabit Ethernet Card

The general procedure for circuit performance measurement with the end-system setup will be as follows:

- Set up end-to-end circuit path
- Verify basic end-to-end circuit connectivity
- Run data tests across end-to-end circuit (test sequence will include multiple flows at varying rates including 1 Gbps, 5 Gbps, and 10 Gbps and also varying MTU sizes)
- Collect circuit performance metrics including loss rate, jitter, and latency for each test run
- Repeat tests with TCP and UDP, collect circuit performance metrics including loss rate, jitter, and latency for each test run. TCP tests will also include measurements on re-transmissions

The specific tests descriptions, data collected, and end-to-end path combinations to be exercised are further detailed in Section 4.

3.3. Measurement Suite

We include a spectrum of measurements ranging from link-level measurements that include special-purpose hardware to application-level end host measurements:

1. Latency: Spirent measurements, ping (ICMP), TCP (rtt) and application-level clinet-server
2. Packet loss: segment level, IP-level and application-level (UDP)
3. Throughput: Iperf, bbcp, UDT, hurricane and others

4. Network Test Plan

This section identifies the various end-to-end network paths that will be tested as part of this plan. Here Figure 2 shows the links available for this testing phase in each of the desired networks (Section 2), with the estimated fiber mileages between the locations is shown in the upper right hand side. Specifically, these links will be used to construct intra-network and inter-network circuits for testing. Moreover, the test equipment will be located at four locations, i.e., Sunnyvale CA (SUNV), Chicago IL (CHIN), Oak Ridge TN (ORNL), and Washington DC (WASH). These four locations provide access to all the networks under test and provide cross-connection capabilities for inter-network testing purposes. Note that the Lawrence Berkeley Lab (LBNL) will be the actual hosting site for the testing equipment at the SUNV location. In addition, there is a 10-Gigabit local connection between LBNL and SUNV.

Figure 2: Network links available for test

Using the notations as defined in Section 2, intra-network and inter-network paths are identified for performance testing as described in the following sub-sections.

4.1. Intra-Network Paths

Table 2 shows the intra-network paths to be tested a part of this plan.

Table 2 Intra-network paths

Circuit Type <i>network[accessframing:dataplane:accessframing]</i>	Circuit Path <i>network[ingressloc:transitnodes:egressloc]</i>
<i>esnet [ethernet:psc:ethernet]</i>	<i>esnet [WASH:NYCM:CHIN]</i>
<i>esnet [ethernet:psc:ethernet]</i>	<i>esnet [WASH:NYCM:CHIN:SUNV]</i>
<i>esnet [ethernet:psc:ethernet]</i>	<i>esnet [CHIN:SUNV]</i>
<i>esnet [ethernet:pscq:ethernet]</i>	<i>esnet [WASH:NYCM:CHIN]</i>
<i>esnet [ethernet:pscq:ethernet]</i>	<i>esnet [WASH:NYCM:CHIN:SUNV]</i>
<i>esnet [ethernet:pscq:ethernet]</i>	<i>esnet [CHIN:SUNV]</i>
<i>abilene [ethernet:psc:ethernet]</i>	<i>abilene [WASH:NYCM:CHIN]</i>
<i>abilene [ethernet:psc:ethernet]</i>	<i>abilene [WASH:NYCM:CHIN:IPLS:KSCY:DNVR:SUNV]</i>
<i>abilene [ethernet:psc:ethernet]</i>	<i>abilene [CHIN:IPLS:KSCY:DNVR:SUNV]</i>
<i>abilene [ethernet:pscq:ethernet]</i>	<i>abilene [WASH:NYCM:CHIN]</i>
<i>abilene [ethernet:pscq:ethernet]</i>	<i>abilene [WASH:NYCM:CHIN:IPLS:KSCY:DNVR:SUNV]</i>
<i>abilene [ethernet:pscq:ethernet]</i>	<i>abilene [CHIN:IPLS:KSCY:DNVR:SUNV]</i>
<i>sdn [ethernet:pscq:ethernet]</i>	<i>sdn [ORNL:WASH]</i>
<i>sdn [ethernet:pscq:ethernet]</i>	<i>sdn [ORNL:WASH:CHIN]</i>
<i>sdn [ethernet:pscq:ethernet]</i>	<i>sdn [WASH:CHIN]</i>
<i>sdn [ethernet:l2sc:ethernet]</i>	<i>sdn [ORNL:WASH]</i>
<i>sdn [ethernet:l2sc:ethernet]</i>	<i>sdn [ORNL:WASH:CHIN]</i>
<i>sdn [ethernet:l2sc:ethernet]</i>	<i>sdn [WASH:CHIN]</i>

Circuit Type <i>network[accessframing:dataplane:accessframing]</i>	Circuit Path <i>network[ingressloc:transitnodes:egressloc]</i>
<i>usn [ethernet:tdm:ethernet]</i>	<i>usn [ORNL:CHIN]</i>
<i>usn [ethernet:tdm:ethernet]</i>	<i>usn [ORNL:CHIN:STTL:SUNV]</i>
<i>usn [ethernet:tdm:ethernet]</i>	<i>usn [CHIN:STTL:SUNV]</i>
<i>usn [ethernet:l2sc:ethernet]</i>	<i>usn [ORNL:CHIN]</i>
<i>usn [ethernet:l2sc:ethernet]</i>	<i>usn [ORNL:CHIN:STTL:SUNV]</i>
<i>usn [ethernet:l2sc:ethernet]</i>	<i>usn [CHIN:STTL:SUNV]</i>
<i>i2net [ethernet:tdm:ethernet]</i>	<i>i2net [WASH:NYCM:CHIN]</i>
<i>i2net [ethernet:l2sc:ethernet]</i>	<i>i2net [WASH:NYCM:CHIN]</i>
<i>i2net [ethernet:lsc:ethernet]</i>	<i>i2net [WASH:NYCM:CHIN]</i>
<i>dragon [ethernet:lsc:ethernet]</i>	<i>dragon [GSFC:CLPK:ARLG:MCLN]</i>

4.2. Inter-Network Paths

Table 3 shows the inter-network paths to be tested a part of this plan

Table 3 Inter-network paths

Circuit Type <i>network[accessframing:dataplane:accessframing]</i>	Circuit Path <i>network[ingressloc:transitnodes:egressloc]</i>
<i>esnet[ethernet:pscq:ethernet]: usn[ethernet:tdm:ethernet]</i>	<i>esnet[WASH:NYCM:CHIN]:usn[CHIN:ORNL]</i>
<i>esnet[ethernet:pscq:ethernet]: usn[ethernet:tdm:ethernet]</i>	<i>esnet[WASH:NYCM:CHIN]:usn[CHIN:STTL:SUNV]</i>
<i>esnet[ethernet:pscq:ethernet]: abilene[ethernet:pscq:ethernet]</i>	<i>esnet[WASH:NYCM:CHIN]:abilene[WASH:NYCM:CHIN:IPLS:KSCY:DNVR:SUNV]</i>
<i>esnet[ethernet:pscq:ethernet]: usn[ethernet:tdm:ethernet]</i>	<i>esnet[SUNV:CHIN]: usn[CHIN:ORNL]</i>
<i>Esnet[Ethernet,pscq,Ethernet] Usn[Ethernet:tdm:Ethernet]</i>	<i>Esnet[SUNV:CHIN]:usn[CHIN:ORNL:SUN]</i>
<i>esnet[ethernet:pscq:ethernet]: i2net[ethernet:tdm:ethernet]</i>	<i>esnet[WASH:NYCM:CHIN]: i2net[CHIN:NYCM:WASH]</i>
<i>esnet[ethernet:pscq:ethernet]:i2net[ethernet:tdm:ethernet]</i>	<i>esnet[SUNV:CHIN]: i2net[CHIN:WASH]</i>
<i>abilene[ethernet:pscq:ethernet]: i2net[ethernet:tdm:ethernet]</i>	<i>abilene[SUNV:DNVR:KSCY:IPLS:CHIN:NYCM:WASH]: i2net[WASH:NYCM:CHIN]</i>
<i>abilene[ethernet:pscq:ethernet]: usn[ethernet:tdm:ethernet]</i>	<i>abilene[WASH:NYCM:CHIN]: usn[CHIN:STTL:SUNV]</i>
<i>usn[ethernet:tdm:ethernet]:i2net[ethernet:tdm:ethernet]:esnet[ethernet:pscq:ethernet]:usn[ethernet:tdm:ethernet]</i>	<i>usn[ORNL:CHIN]:i2net[CHIN:WASH]:esnet[WASH:CHIN]:usn[CHIN:STTL:SUNV]</i>
<i>esnet[ethernet:pscq:ethernet]: abilene[ethernet:pscq:ethernet]: i2net[ethernet:tdm:ethernet]: usn[ethernet:tdm:ethernet]</i>	<i>esnet[WASH:NYCM:CHIN:SUNV]: abilene[SUNV:DNVR:KSCY:IPLS:CHIN:NYCM:WASH]:i2net[WASH:NYCM:CHIN]: usn[CHIN:STTL:SUNV]</i>

As an example, Figure 3 illustrates the configuration of an inter-network path which spans USN, I2 Network, and ESNet. While this is not necessarily a path that might be provisioned in direct support of an eScience application, is it one that may be considered as an extreme testcase (i.e., in terms of distance and inter-network transitions).

Figure 3: Example of an inter-network path traversing across USN, I2 Network, and ESNet

The formal description of the extended Inter-Network path show in Figure 4 is as follows:

Circuit type: *usn [ethernet:tdm:ethernet] : i2net [ethernet:tdm:ethernet] :
esnet [ethernet:pscq:ethernet] : usn [ethernet:tdm:ethernet]*

Circuit path: *usn [ORNL:CHIN] : i2net [CHIN:WASH] : esnet [WASH:CHIN] :
usn [CHIN:STTL:SUNV]*

This inter-network path has four segments which are concatenated at three cross-layer mapping points. Namely, the mapping between USN and I2 Network is performed by the Ciena multi-service CoreDirector platform at CHIN using the Ethernet-over-SONET mapping feature. Meanwhile the mapping between I2 Network and ESNet is performed by the Juniper router at WASH and CHIN using the circuit cross-connect (CCC) feature. Finally, the mapping between ESNet and USN is performed via sequential back-to-back Juniper router CCC and Ciena Ethernet-over-SONET mappings at CHIN. The configuration of other paths in Table 3 could be described in a similar manner.

4.3. Test Sequence

The above-detailed data plane performance tests will be conducted for each of the paths described in Sections 4.1 and 4.2. Namely, each path under testing will have a raw capacity of up to 10 Gbps. The following steps will be performed for each test.

- Provision a 10-Gigabit path for end-to-end Ethernet VLAN service
- Connect two Spirent AX/4000 to the ingress and egress ports at both ends of the tested path
- Generate a single flow of Ethernet VLAN traffic via the Spirent AX/4000 for each of the sub tests as shown in Table 4
- Run each sub-test continuously for 30 minutes and collect performance statistics
- Disconnect the Spirent AX/4000
- Run the end-system tests as described below
- Tear down the tested path and disconnect test equipment

Table 4: Performance tests using Spirent test equipment

Sub Test	Line Rate	Ethernet VLAN Payload	Data Rate (packets/sec)
1-GigE w/ short IP packets	1 Gbps	Fixed 512-byte IP	10,523
1-GigE w/ long IP packets	1 Gbps	Fixed 9,000-byte IP	10,327
5-GigE w/ short IP packets	5 Gbps	Fixed 512-byte IP	907,617
5-GigE w/ long IP packets	5 Gbps	Fixed 9,000-byte IP	51,633
*10-GigE w/ short IP packets	10 Gbps	Fixed 512-byte IP	1,815,234
*10-GigE w/ long IP packets	10 Gbps	Fixed 9,000-byte IP	103,267
*Some network link will only allow a maximum of 5 Gbps to be provisioned. This is in order to protect production traffic. These links will be noted as part of the data collection.			

Following the completion of all tests using the Spirent gear, the following end-system tests will be run (using the Linux workstation setup, Section 3.2). The following steps will be performed for each test.

- Provision a 10-Gigabit path for end-to-end Ethernet VLAN service
- Connect two end-system test equipment to the ingress and egress ports at both ends of the path
- Generate the data flows via the end system for each of the sub tests as shown in Table 4
- Run each sub-test continuously for 30 minutes and collect performance statistics
- Tear down the tested path and disconnect the test equipment

Table 5: Performance Tests using Linux End System**

Sub Test	Line Rate	Ethernet VLAN Payload	protocol
1-GigE w/ short IP packets	1 Gbps	Fixed 512-byte IP	UDP, TCP
1-GigE w/ long IP packets	1 Gbps	Fixed 9,000-byte IP	UDP, TCP
5-GigE w/ short IP packets	5 Gbps	Fixed 512-byte IP	UDP, TCP
5-GigE w/ long IP packets	5 Gbps	Fixed 9,000-byte IP	UDP, TCP
*10-GigE w/ short IP packets	10 Gbps	Fixed 512-byte IP	UDP, TCP
*10-GigE w/ long IP packets	10 Gbps	Fixed 9,000-byte IP	UDP, TCP
*Some network link will only allow a maximum of 5 Gbps to be provisioned. This is in order to protect production traffic. These links will be noted as part of the data collection.			
**These tests will use NUTTCP for memory to memory testing. These end systems tests will not be run on all data paths provisioned. A specific set of circuits will be identified for these tests.			

5. Analysis Framework

We utilize the measurement collected over various connections to carry out analysis to compare the performance of various individual technologies and their combinations. We will utilize segmented regression method to correlate the connection lengths with parameters including throughput, loss rate, and jitter. In each case, for each type of connection, we plot the measurements as a function of connection length l . We then utilized regression modeling in different ranges of connection lengths to compute the mean value $\bar{D}(l)$ as a function of l . Then we partition the measurements that are above and below $\bar{D}(l)$ and compute upper regression $\bar{D}_U(l)$ and lower regression $\bar{D}_L(l)$, respectively. We can compute the parameter band $\bar{B}_{UL}(l) = \bar{D}_U(l) - \bar{D}_L(l)$ that characterizes the variability of the measurement as illustrated in Figure for jitter measurements.

Figure4. Illustration of regression band computation for jitter.

We now briefly describe the jitter analysis performed on usn [Ethernet;tdm:Ethernet] on 1Gbps connections of the following lengths and rtt's shown in the left plot of Figure 5. We show measurements in the right plot of Figure 5, where bottom and top correspond to ping and client-server measurements, respectively and the middle corresponds to tcpmon measurements.

Figure 5. Connection lengths and ping rtt's for 1Gbps connections on USN shown in the left and measurement suite shown in the right.

Based on the regressions, various parameters can be interpolated or extrapolated at connection lengths at which measurements are not available. Once these values are known the measurement sets collected at two different connection lengths can be aligned and compared as illustrated in Figure 6.

Figure 6. Normalization framework for comparing measurements of different types collected at different connection lengths.

6. Modeling and Simulation Development Plan

Overall, the network test plan outlined in Section 4 will yield many detailed empirical results on the performance of different circuit types and combinations. In order to further extend this work, however, additional network simulation activities will also be conducted. Specifically, detailed data plane models will be developed for all network element types (e.g., Ethernet switches, MPLS routers, SONET cross-connects, etc) and end systems (e.g., test systems, work-stations, test applications). One of the key goals here will be to first verify and corroborate empirical findings in network simulation environments. Subsequently, the broader goal will be to extrapolate and test broader test scenarios via the simulation environment, i.e., ones that may be very difficult to recreate in the real-world test networks.

All simulation development will be done using the *OPNET ModelerTM* development environment. This package features a hierarchical design approach (subnetwork-network-node-process) and provides a solid graphical user-interface (GUI) interface for ease of use, Figure 4. In addition, the tool provides a comprehensive C/C++ backend for coding/development of very specialized network element and end-system types. Now it is well-known that *OPNET ModelerTM* provides many existing pre-programmed models for IP, Ethernet, ATM, frame relay, and other nodes. Nevertheless, SONET/SDH and optical WDM switching element models are lacking. Therefore it is difficult to directly use the existing *OPNET ModelerTM* suite to model the specific data plane tests specified in the network test plan (Section 4). Alternatively, modifying existing *OPNET ModelerTM* code blocks is very cumbersome. Therefore a more effective “clean-slate” approach is pursued in which comprehensive data plane nodes will be specifically developed for all network element and end-systems types of interest to DOE researchers. This phase will make extensive re-use of the existing models developed by some of the project personnel to model multi-domain networks. Some details are provided.

Figure 4: Sample data plane simulation models developed in *OPNET Modeler*TM

6.1. Network Element Models

The network element models will develop two broad node types, namely packet-switching and circuit-switching nodes. Specifically, packet-switching nodes will be designed around a generic core node model that can be tailored to represent any desired QoS-capable Layer 2 (L2SC) or Layer 3 (PSC) device. Here the core node will implement processes for input/output link buffer management and scheduling along with core-switching fabric functionalities. This generic model will then be tailored by wrapping around more specific “packet-layer” encapsulations and switching functionalities, e.g., IP packets and routing lookup tables, MPLS shim headers and label switching tables, Ethernet packets and switching tables, etc. All of these node models will allow users to directly set related data plane parameters, e.g., buffer sizes, scheduling policies, packet buffering/discard policies, switching delays, input-output table associations, etc.

Meanwhile, the circuit-switching nodes will implement data plane capabilities of SONET/SDH (TDM) and optical DWDM (LSC) switching nodes. As these elements do not perform any “data-packet” inspection per say, related models will be simpler as delays will largely comprise minimal framing and bypass switching times. These figures will hold regardless of the circuit load being carried across the switch. Overall, the developed node models will allow users to build any arbitrary network test topology under consideration, e.g., portions of ESNet, USN, SDN, Abilene, DRAGON.

6.2. End-System Models

Given the numerous testing tools to be used in the empirical test-plan, it is vital to develop commensurate models in the simulation environment. Along these lines, a host of end-system node types will be developed for data plane testing/monitoring purposes. Specifically these include work-stations running various specialized applications such as UDP and TCP file-transfer, ICMP ping, TCP MON applications, and possibly even iperf and nuttcp. In all these designs end-users will be able to specify a full range of test parameters, e.g., file sizes, timer values, port speeds, maximum segment sizes, etc. Detailed end-to-end and node-level statistics will also be collected, particularly on packet delay, jitter, and loss. In addition, end-users will have the capability to develop and test any network test scenario by simple drag-and-drop

operations, i.e., interconnect arbitrary work-station/test-application combination with ports on the various network element models (Section 5.1).

6.3. Simulation Test Plan

The simulation test plan will comprise of two key phases. The first part will mirror the various test scenarios specified in Section 4, particularly inter-domain paths. Namely, detailed topologies will be built to reflect the real-world networks under test, with the end goal of gauging the accuracy of the developed simulation models. Pending successful correlation and findings here, the second part of the simulation test plan will then look at testing more complex inter-domain path scenarios. In particular these tests will focus on the inclusion of multiple concurrent paths with various intersecting links in the Layer 2 and Layer 3 network domains. These types of scenarios are expected to yield increased delay/jitter variations. The goal here will be to extrapolate and generalize the findings from the empirical test plan.

7. Summary

This document described a plan to test a diversified set of data paths within and across the ESnet, SDN, USN, Abilene, Internet2 I2 Network, and DRAGON networks. By measuring the latency, loss and jitter of these paths, we hope to gain better understanding of the performance characteristics of long-distance high-capacity paths in a hybrid multi-layer networking environment and compare different networking technologies as well as different combinations/concatenations of these technologies. What has been described in this document is the initial set of tests (Phase I) in our project. Based on the results from these tests, further tests will be conducted for more precise observation and comparison of a new selection of data paths. The longer term goal is to formalize a common service definition which will properly specify and support these parameters on a user-selectable basis. The testing results will also help determine how best to provision these common service requirements in the most efficient manner. To augment the data collection and analysis, a data plane modeling and simulation capability is also being developed. These simulation models will be used to further evaluate the data collected as well as form the basis for hybrid network design, analysis, and optimization.

APPENDIX A: Network and Test Equipment Configuration

This Appendix will contain details on the network configurations, test equipment attachment points, and associated diagrams. This will include the following information:

- Test equipment parts list and configuration
- Where/how to install test equipment
- Where/how to connect test equipment to individual network elements. This includes identification of physical ports and any configuration changes required on individual network elements.
- Procedures for provisioning of circuits. This will include the provision of circuits across the individual networks as well as procedures for the concatenation of circuits between two networks.

APPENDIX B: Test Procedures and Data Collection

This appendix will contain the detailed instructions on what tests to run, how to run them, and what data to collect. Forms will be included to record data.

APPENDIX C: Segmented Regression Method

This appendix will contain detailed formulae for regression analysis.

Hybrid Multilayer Network Data Planes: Test Configuration

v6.2

1.1. Intra-Network Paths

The below table shows the intra-network paths to be tested as part of this plan.

Intra-network paths

#	Circuit Type <i>network[accessframing:dataplane:accessframing]</i>	Circuit Path <i>network[ingressloc:transitnodes:egressloc]</i>	Interface Speed
1.	<i>usn [ethernet:tdm:ethernet]</i>	<i>usn [ORNL-SPIRENT:ORNL-CD:CHIN-CD: CHIN-CD:ORNL-CD:ORNL-SPIRENT]</i>	1G
2.	<i>usn [ethernet:tdm:ethernet]</i>	<i>usn [ORNL-SPIRENT:ORNL-CD:CHIN-CD: STTL-CD:STTL-CD:CHIN-CD:ORNL-CD:ORNL-SPIRENT]</i>	1G
3.	<i>usn [ethernet:tdm:ethernet]</i>	<i>usn [ORNL-SPIRENT:ORNL-CD:CHIN-CD: STTL-CD:SUNV-CD:SUNV-CD:STTL-CD:CHIN-CD:ORNL-CD:ORNL-SPIRENT]</i>	1G
4.	<i>usn[ethernet:l2sc:ethernet]</i>	<i>usn [ORNL-SPIRENT:ORNL-E300:CHIN-E300: CHIN-E300:ORNL-E300:ORNL-SPIRENT]</i>	1G
5.	<i>usn[ethernet:l2sc:ethernet]</i>	<i>usn [ORNL-SPIRENT:ORNL-E300:CHIN-E300: CHIN-E300:SUNV-E300:SUNV-E300:ORNL-E300:ORNL-SPIRENT]</i>	1G
6.	<i>usn[ethernet:l2sc:ethernet]</i>	<i>usn [ORNL-SPIRENT:ORNL-E300:CHIN-E300: CHIN-E300:CHIN-SPIRENT]</i>	1G,10G
7.	<i>usn[ethernet:l2sc:ethernet]</i>	<i>usn [ORNL-SPIRENT:ORNL-E300:CHIN-E300: CHIN-E300:SUNV-E300:SUNV-E300:CHIN-E300:CHIN-SPIRENT]</i>	1G,10G
8.	<i>esnet[ethernet:psc:ethernet]</i>	<i>esnet[CHIN-SPIRENT:CHIN-E300:CHIN-RTR:SUNV-RTR:SUNV-E300:SUNV-RTR:CHIN-RTR:CHIN-E300:CHIN-SPIRENT]</i>	1G
9.	<i>esnet[ethernet:psc:ethernet]</i>	<i>esnet[CHIN-SPIRENT:CHIN-E300:CHIN-RTR:SUNV-RTR:SUNV-E300:ORNL-E300:ORNL-SPIRENT]</i>	1G
10.	<i>i2hopi[ethernet:l2sc:ethernet]</i>	<i>i2hopi[MCLN-SPIRENT:MCLN-E600:CHIN-E600:CHIN-SPIRENT]</i>	1G,10G
11.	<i>abilene[ethernet:psc:ethernet]</i>	<i>abilene[MCLN-SPIRENT:MCLN-E600:MCLN-RTR:CHIN-RTR:CHIN-E600:CHIN-SPIRENT]</i>	1G
12.	<i>dragon[ethernet:l2sc:ethernet]</i>	<i>dragon[ARLG-SPIRENT:ARLG-RAPTOR:MCLN-RAPTOR:MCLN-RAPTOR:ARLG-RAPTOR:ARLG-SPIRENT]</i>	1G
13.	<i>dragon[ethernet:l2sc:ethernet]</i>	<i>dragon[ARLG-SPIRENT:ARLG-RAPTOR:MCLN-RAPTOR:MCLN-SPIRENT]</i>	1G,10G
14.	<i>dragon[ethernet:lsc:ethernet]</i>	<i>dragon[ARLG-SPIRENT:ARLG-WDM:MCLN-WDM:MCLN-WDM:ARLG-WDM:ARLG-SPIRENT]</i>	1G
15.	<i>dragon[ethernet:lsc:ethernet]</i>	<i>dragon[ARLG-SPIRENT:ARLG-WDM: MCLN-WDM:MCLN-SPIRENT]</i>	1G,10G
16.	<i>dragon[ethernet:lsc:ethernet]</i>	<i>dragon[ARLG-SPIRENT:ARLG-WDM:DCNE-WDM:CLPK-WDM:DCGW-WDM:MCLN-WDM:ARLG-WDM:ARLG-SPIRENT]</i>	1G,10G

1.2. Inter-Network Paths

The Table below shows the inter-network paths to be tested a part of this plan

Inter-network paths			
#	Circuit Type <i>network[accessframing:dataplane:accessframing]</i>	Circuit Path <i>network[ingressloc:transitnodes:egressloc]</i>	Interface Speed
17.	<i>usn [ethernet:tdm:ethernet]: esnet[ethernet:psc:ethernet]</i>	<i>usn [ORNL-SPIRENT:ORNL- CD:CHIN-CD]:esnet[CHIN- E300:CHIN-RTR:SUNV-RTR:SUNV- E300:SUNV-RTR:CHIN-RTR:CHIN- E300:CHIN-SPIRENT]</i>	1G
18.	<i>usn [ethernet:tdm:ethernet]: i2hopi[ethernet:l2sc:ethernet]</i>	<i>usn [ORNL-SPIRENT:ORNL- CD:CHIN-CD]: i2hopi[CHIN- E600:MCLN-E600:MCLN-SPIRENT]</i>	1G,10G
19.	<i>usn [ethernet:tdm:ethernet]: i2hopi[ethernet:l2sc:ethernet]:dragon[ethernet:lsc:ethernet]</i>	<i>usn [ORNL-SPIRENT:ORNL- CD:CHIN-CD]: i2hopi[CHIN- E600:MCLN-E600]:dragon[MCLN- WDM:ARLG-WDM:ARLG-SPIRENT]</i>	1G,10G
20.	<i>usn [ethernet:tdm:ethernet]: i2hopi[ethernet:l2sc:ethernet]:abilene[ethernet:psc:ethernet]</i>	<i>usn [ORNL-SPIRENT:ORNL- CD:CHIN-CD]: i2hopi[CHIN- E600]:abilene[CHIN-RTR:MCLN- RTR:MCLN-E600:MCLN-SPIRENT]</i>	1G
21.	<i>usn [ethernet:tdm:ethernet]: i2hopi[ethernet:l2sc:ethernet]:abilene[ethernet:psc:ethernet]:dragon[ethernet:lsc:ethernet]</i>	<i>usn [ORNL-SPIRENT:ORNL- CD:CHIN-CD]: i2hopi[CHIN- E600]:abilene[CHIN-RTR:MCLN- RTR:MCLN-E600]:dragon[MCLN- WDM:ARLG-WDM:ARLG-SPIRENT]</i>	1G
22.	<i>i2hopi[ethernet:l2sc:ethernet]:abilene[ethernet:psc:ethernet]</i>	<i>i2hopi[CHIN-SPIRENT:CHIN- E600]:abilene[CHIN-RTR:MCLN- RTR]:i2hopi[MCLN-E600:MCLN- SPIRENT]</i>	1G
23.	<i>i2hopi[ethernet:l2sc:ethernet]:abilene[ethernet:psc:ethernet]:dragon[ethernet:lsc:ethernet]</i>	<i>i2hopi[CHIN-SPIRENT:CHIN- E600]abilene[CHIN-RTR:MCLN-RTR: MCLN-E600]:dragon[MCLN- WDM:ARLG-WDM: ARLG-SPIRENT]</i>	1G
24.	<i>i2hopi[ethernet:l2sc:ethernet]:esnet[ethernet:psc:ethernet]</i>	<i>i2hopi[MCLN-SPIRENT:MCLN- E600:CHIN-E600]:esnet[CHIN- E300:CHIN-RTR:SUNV-RTR:SUNV- E300:SUNV-E300:SUNV-RTR:CHIN- RTR:CHIN-E300:CHIN-SPIRENT]</i>	1G
25.	<i>i2hopi[ethernet:l2sc:ethernet]:abilene[ethernet:psc:ethernet]:dragon[ethernet:l2sc:ethernet]</i>	<i>i2hopi[CHIN-SPIRENT:CHIN- E600]:abilene[CHIN-RTR:MCLN- RTR:MCLN-E600]:dragon[MCLN- WDM-ARLG-WDM:ARLG-SPIRENT]</i>	1G
26.	<i>i2hopi[ethernet:l2sc:ethernet]:dragon[ethernet:lsc:ethernet]</i>	<i>i2hopi[CHIN-SPIRENT:CHIN- E600:MCLN-E600:]dragon[MCLN- WDM:ARLG-WDM: ARLG-SPIRENT]</i>	1G,10G

1.3. Data Collection

The Table below shows data to be collected for each network configuration.

1 Gigabit Interfaces				
MTU (bytes)	Data Rate (Mbps)	Packet Rate (Pkt/s)	Run Time (sec)	Total Packets
64	100	195313	300	117187500
512	500	122070	300	73242188
8000	800	12500	300	7500000

10 Gigabit Interfaces				
MTU (bytes)	Data Rate (Gbps)	Packet Rate (Pkt/s)	Run Time (sec)	Total Packets
64	1	1953125	300	1171875000
512	5	1220703	300	732421875
8000	8	125000	300	75000000

For each test run, the following data will be collected:

- average datarate
- total packet loss
- average latency
- jitter profile (histogram)
- transfer delay (histogram)

In addition some shorter runs will be made to collect sequenced data. These test are limited in duration by the capture buffer size on test equipment. These tests will be run at the same time as the above tests. As a result, they will be a packet trace of a subset of the above data.

1 Gigabit Interfaces - short duration run		
MTU (bytes)	Data Rate (Mbps)	Packet Rate (Pkt/s)
64	100	195313
512	500	122070
8000	800	12500

1 Gigabit Interfaces - short duration run		
MTU (bytes)	Data Rate (Gbps)	Packet Rate (Pkt/s)
64	1	1953125
512	5	1220703
8000	8	125000

For each test run, the following data will be collected:

- average datarate

- total packet loss
- average latency
- jitter profile (histogram)
- packet sequence data

The data described above will be collected under two different background network loading configurations

- "light background traffic"
- "heavy background traffic"

Since we cannot control all the network configurations and background traffic profiles, the exact conditions associated with the above two categories will be determined and documented at test time. In general, the "light background traffic" case will be represent the base background traffic in place due to other current users on the network. The "heavy background traffic" will include traffic being injected into the network by the testers as part of the test program. The ability (and permission) to do this will vary by network. The objective will be to introduce the maximum amount of background traffic on the netowrk as possible. Since we are testing on real networks, we are somewhat constrained in what we can do in this regard. However, a key requirment will be to document to the maximum extent possible, the current network conditions during data collection.

Additional Notes:

For *[ethernet:tdm:ethernet]* flows:

-1 Gigabit Ethernet Interfaces: the Ethernet port on TDM box is mapped into 21 STS-1s

-10 Gigabit Ethernet Interfaces: the Ethernet port on TDM box is mapped into 171 STS-1s

For *[ethernet:l2sc:ethernet]* flows:

-1 Gigabit Ethernet Interfaces: An Ethernet VLAN with 1 Gigabit/s ingress policing and egress rate limiting is configured on all Ethernet switches in path.

-10 Gigabit Ethernet Interfaces: An Ethernet VLAN with 9 Gigabit/s ingress policing and egress rate limiting is configured on all Ethernet switches in path.

-For all l2sc flows which also take an underlying TDM circuit between ethernet switches, the underlying TDM circuits should map 21 STS-1s for 1 Gigabit connections, and 192 STS-1s for 10 Gigabit connections between switches.

For *[ethernet:lsc:ethernet]* flows:

-1 Gigabit Ethernet Interfaces: the Ethernet port on LSC box is mapped to a 1 Gigabit lambda.

-10 Gigabit Ethernet Interfaces: the Ethernet port on LSC box is mapped to a 1 Gigabit lambda.

-For all l2sc flows which also take an underlying lsc circuit between ethernet switches, the underlying lsc circuits should map a 1 Gigabit lambda for 1 Gigabit connections, and a 10 gigabit lambda for 10 Gigabit connections between switches.

For *[ethernet:psc:ethernet]* flows:

-1 Gigabit Ethernet Interfaces: the Ethernet port on router is mapped into a VLAN/LSP which 1 Gigabit/s QoS applied

1.3.1. Data Folder Naming

Data Collection and File Naming

The following will be saved from each network element:

- config file
- port(s) buffer size and configuration,
- traffic loading at time of test

Test numbering/naming is as follows:

- test#-mtu@-bw\$
 - test#: test number from above tables(number indicates multiple runs of same test type)
 - mtu@: 64|512|8000 corresponding to mtu size
 - bw\$: 100|500|800|1000|5000|8000 Mbps

As an example, here is the test designation for Test 12:

12a-mtu64-bw100
 12a-mtu64-bw500
 12a-mtu64-bw800
 12a-mtu512-bw100
 12a-mtu512-bw500
 12a-mtu512-bw800
 12a-mtu8000-bw100
 12a-mtu8000-bw500
 12a-mtu8000-bw800

Spirent AX4000 Config Files are named as follows:

- mtu@-bw\$.ipt
 - @: 64|512|8000 corresponding to mtu size
 - bw\$: 100|500|800|1000|5000|8000 Mbps
- vlan*.ipp
 - &: corresponds to the vlan tag used (0 means not vlan tag)
 - this file also specifies src and dest IP address which are always 1.1.1.1 and 1.1.1.2

The following Spirent AX4000 config files are identified:

mtu64-bw100.ipt
 mtu64-bw500.ipt
 mtu64-bw800.ipt
 mtu64-bw1000.ipt
 mtu64-bw5000.ipt
 mtu64-bw8000.ipt

mtu512-bw100.ipt
 mtu512-bw500.ipt
 mtu512-bw800.ipt
 mtu512-bw1000.ipt
 mtu512-bw5000.ipt
 mtu512-bw8000.ipt

mtu8000-bw100.ipt
 mtu8000-bw500.ipt
 mtu8000-bw800.ipt
 mtu8000-bw1000.ipt
 mtu8000-bw5000.ipt
 mtu8000-bw8000.ipt

vlan0.ipp
 vlan3120.ipp
 vlan3121.ipp

vlan616.ipp

1.3.2. Test Sequence

The following is the general sequence of steps for each test run. All saved information is stored in a folder with a name per the number per above section.

1. Load proper .ipt and .ipp files, save the following
 - .xls file with test name and path details
2. Run Interarrival Histogram during 1-2 minute interval, save the following:
 - interarrival-histogram.text
 - interarrival-histogram.jpg
3. Run Packet capture starting at 3 minute mark
 - settings of: 128Kbyte captures size, 2Kbyte trigger, selected substream of 1.1.1.1
 - capture.csi
 - capture.enc
 - capture-list-packetmode.txt
 - capture-detail-packetmode.txt
 - capture-list-decodemode.txt
 - capture-detail-decodemode.txt
4. Run Latency Histogram during 4-5 minute interval, save the following:
 - transferdelay-histogram.text
 - transferdelay-histogram.jpg
5. Screen capture of overall stream statistics at end of run (5 minute mark)
 - statistics.rtf

1.4. Schedule

Schedule for testing is as follows:

Dates	Activities
Feb 26-Mar 2	-IntraNetwork Tests for USN, DRAGON, HOPI, ESnet. Connect up USN to HOPI patch in Chicago
Mar 5-9	-InterNetwork Tests for USN, DRAGON, HOPI, ESnet
Mar 12-16	- Finish Testing
Mar 19-23	- Finish Testing
Mar 26	-Ship back AX4000s on March 26 for March 27 arrival

Appendix D

Hybrid-MLN Presentations

Multi-Layer Networking

An Architecture Framework

Tom Lehman (USC/ISI)

Chin Guok (ESnet)

Brian Tierney (ESnet)

Nasir Ghani (UNM)

Multi-Layer Network Architecture

- This work is an attempt to capture ongoing discussions regarding multi-layer networks within the DOE community (and beyond)
- The desire is to develop a framework for discussion
 - not intended to define or require anything specific regarding implementations or designs
- Intended audience is researchers and network system implementers in DOE

Multi-Layer Network Architecture

- Network architectures and implementations are on the edge of a transformation from a single-layer to a multi-layer paradigm.
- This is in the context of "network service provision and control"
 - physical construction has always been multi-layer
- In many ways, this represents an evolution from IP QoS, to MPLS, to "GMPLS" (below routers) network control and provisioning.

Multi-Layer Network Architecture

- Additional benefits of the multi-layer paradigm include:
 - higher efficiency (create bandwidth where needed)
 - lower costs
 - bandwidth is cheaper at the lower layer (e.g. cut-through routing)
 - service flexibility and network agility
 - can provide more than just IP service
 - better service guarantees
 - use the transport technology that best meets the request constraints (e.g. SONET, WDM for low jitter, etc)

Multi-Layer Network Architecture

- As we address the complexities of integrated "network service provision and control " across multiple network layers:
 - a common set of terminology/constructs will facilitate discussions, designs, and implementations
- Another goal of this document to provide a framework, context, and definition of where more detailed research and development efforts are needed.

Data Plane Descriptions

- For this architecture definition we identify the dataplane options in terms of “technology regions”.
- A “technology region” is a set of network elements which are grouped together and utilize the same dataplane “technology type”.
- The technology types are defined using the standard Generalized Multi-Protocol Label Switching (GMPLS) nomenclature:
 - Packet Switching Capable (PSC) layer
 - Layer-2 Switching Capable (L2SC) layer
 - Time Division Multiplexing (TDM) layer
 - Lambda Switching Capable (LSC) layer
 - Fiber-Switch Capable (FSC)

Data Plane Descriptions

- Additionally, we can associate these technology types with the more common layer terminology:
 - Layer 3 for PSC (IP Routing)
 - Layer 2.5 for PSC (MPLS)
 - Layer 2 for L2SC (often Ethernet)
 - Layer 1.5 for TDM (often SONET/SDH)
 - Layer 1 for LSC (often WDM switch elements)
 - Layer 0 for FSC (often port switching devices based on optical or mechanical technologies)
- DataPlane - A set of network elements which receive, send, and switch the network data
 - For this architecture work we assume each technology region is of a single technology type. In reality a technology region may have multiple technology types.

Data Plane Technology Types/Regions

Layer 3
(PSC)

Layer1.5
(TDM)

Layer 3/Layer2.5
(PSC with MPLS)

Layer1
(LSC)

Layer2
(L2SC)

Layer0
(FSC)

Many Multi-Layer Topology Options

Current Network Provisioning

- Horizontal Inter-Domain Provisioning
- Single Layer Intra-Domain
- using systems like OSCARS, DRAGON

Vertical Intra-Domain is Desired

- Provision services at lower layer to create a topology element (link between routers) at the higher layer
- Subsequently provision any remaining bandwidth at the higher level
 - may provision a 10Gbps LSC link in response to a request for 5Gbps immediate need
 - remaining 5Gbps available for subsequent service requests

Vertical Inter-Domain Even Better

- Lower layer service provisioning may cross a domain boundary
- Networks may peer a multiple layers and flexibly provision services

Multi-Layer Architecture Capability Planes

CapabilityPlanes - Graphical View

CapabilityPlanes - Functional Layered View

DataPlane Layer Centric View of Capabilities

Multi-Layer Architecture

Capability Planes

- ***CapabilityPlanes*** are defined to capture the key areas required to manage and control a DataPlane layer.
- **ControlPlane**
 - routing, path computation, and signaling
 - maintaining topology information and configuring network elements in terms of data ingress, egress, and switching operations
 - one of two CapabilityPlanes which directly interacts with the DataPlane

Multi-Layer Architecture

Capability Planes

- ServicePlane

- set of systems and processes that are geared towards providing services to users and maintaining state on those services
- will generally rely on the functions of the ControlPlane and/or ManagementPlane to effect actual changes on the DataPlane
- will typically maintain databases on current and future service instantiations and coordinate associated workflow processes

Multi-Layer Architecture Capability Planes

- **ManagementPlane**

- monitor, manage, and troubleshoot the network
- includes functions to support user services as well as network maintenance, upgrades, and reconfigurations
- responsible for collecting data and monitoring of the network
- may also include capabilities to configure the network elements with the support of the control plane or via independent actions
- one of two CapabilityPlanes which directly interacts with the DataPlane.

Multi-Layer Architecture Capability Planes

- **AAPlane** (Authentication and Authorization)
 - responsible for the mechanisms which allow the other planes to identify and authenticate users and receive associated policy information
- **ApplicationPlane**
 - provides higher level functions which will generally be tailored for domain specific purposes.
 - relies on the capabilities offered by one or more ServicePlanes to accomplish its objectives
 - boundary between the ApplicationPlane and ServicePlane is the network demarcation point

Multi-Layer Architecture Capability Planes

- For Each CapabilityPlane we define the following:
 - Functions
 - Function Interfaces
 - Layer Unique Considerations
 - Security Considerations
- For Example:

WorkFlow Development

- Single Domain Circuit Request

- Multi-Domain Circuit Request - InterDomain

WorkFlow Development

- Monitoring and Troubleshooting Workflow

- Many workflows possible including multi-layer integrated vertical provisioning scenarios

Multi-Layer Architectures and Service Concepts

- Multi-Layer Networking - Vertical
- Multi-Layer Networking – Horizontal
- Multi -Layer Networking – Combined
- Multi-Layer Networking – InterDomain
- Hybrid Networking (Multi-Layer Traffic Engineering and Traffic Grooming)
- Nested Capability Planes
- Discussed in architecture document

Architecture Mapping to Existing Implementations

- OSCARS and DRAGON are example of deployed implementations
 - Initial capability sets, so do not address all the areas in MLN architecture
- But we can map the current capabilities of these implementations to the MLN architecture

Capabilities Addressed by OSCARS

- Another slide set with more details on OSCARS is available as well:
 - ESnet OSCARS (oscars-escc-20090723.ppt)

Capabilities Addressed by DRAGON

Multi-Layer Network Architecture

- This slide set presents an overview of the following document:
 - Multi-Layer Networking, An Architectural Framework
- This document is available on ESCC meeting web site
- Reviews are requested, please send comments to:
 - Tom Lehman (tlehman@isi.edu)
 - Chin Guok (chin@es.net)

Future Work

- Ask for review and feedback on MLN Architecture document from multiple communities
- MLN Architecture mostly applies to wide area networks
- Similar types of architecture work planned for Site/Local Area/End System infrastructures
 - Phil DeMar to discuss

Network Convergence Looking Forward

*Networking Research Challenges Workshop
September 28, 2008
Seattle, Washington*

Tom Lehman
Information Sciences Institute East
University of Southern California

Overview

- Next Generation and Emerging Network Architectures - Hybrid Networks
- Hybrid Networks - Where are today?
- Relationship/Differences between R&E and Commercial network spaces
- Vision for R&E network research and development moving forward

Emerging Network Architectures

- There has been significant activity and progress in development and deployment of hybrid network architectures over the last couple of years
- Network technologies, elements, features continue to evolve
- R&E communities use and deployment of these capabilities continues to evolve

Multi-Domain, Multi-Layer Hybrid Networks

- Hybrid networks are intended to provide a flexible mix of IP routed service and dedicated capacity “circuits”
- The “**Multi-Layer**” is meant to identify several items regarding how hybrid networks may be built. In this context it includes the following:
 - **Multi-Technology** - MPLS, Ethernet, Ethernet PBB-TE, SONET, NG-SONET, T-MPLS, WDM
 - **Multi-Level** - domains or network regions may operate in different routing areas/regions, and maybe be presented in an abstracted manner across area/region boundaries
- **Multi-Domain** indicates that we want to allow hybrid network service instantiation across multiple domains
- But there are other “**Multi-**” parameters as well

Multi-Domain, Multi-Layer Hybrid Networks

- **Multi-Service:** This refers to the client experience when they connect to the edge of a dynamic network. Typical service definitions are characterized by the combination of the physical port type (e.g. Ethernet, SONET/SDH, Fibre Channel, etc), the network transport instance (e.g. IP Routed, Ethernet VLAN, SONET), and performance characteristics (e.g. bandwidth, QoS specifications).
- **Multi-Vendor:** This is a reflection that advanced networks will be constructed based on technologies from multiple vendors. A key challenge will be to develop technologies and mechanisms which allow integrated control and service provisioning in this multi-vendor environment.
- **Multi-Policy:** Access to and use of various network components, regions, or topologies may vary by user and/or community due to provider policies.
- **Multi-X** environment

Multi-Level, Multi-Technology, Multi-Vendor Infrastructures

- Multiple Options, most will have vendor proprietary control and management mechanisms which only work across single vendor regions

Multi-Level, Multi-Technology, Multi-Vendor – Network Virtualization

- Network Virtualization and Topology Building in Multi-Level, Multi-Technology, Multi-Vendor Infrastructures

Multi-Level, Multi-Technology, Multi-Vendor Infrastructures

- Current dynamic provisioning environment can be described as:

- **Static Topology, Dynamic Provisioning**

- Next we want to enable:

- **Dynamic Topology, Dynamic Provisioning**

Multi-X, Multi-Domain Control Plane

What can we do today?

- Dynamic provision of end to end (circuits) across multiple domains.
- Specify a few basic parameters regarding a single circuit request: edge technology/configuration, bandwidth, end points, domain sequence, specific start/stop times
- There are multiple projects/efforts/activities around the world working on these types of issues from a multi-domain perspective

InterDomain Messaging Agreements in place

- Web Service Definitions
- Originally developed in DICE
 - Dante, Internet2, CANARIE, ESNNet
 - now includes other organizations as well
- wsdl - web service definition of message types and formats
- xsd – definition of schemas used for network topology descriptions and path definitions

InterDomain Protocol Standardization Activities

- Standardization process and increasing community involvement continues
- GLIF
 - Control Plane Subgroup working on normalizing between various interdomain protocols (IDCP, G-Lambda GNS-WSI, Phosphorus API)
- Open Grid Forum (OGF)
 - Network Service Interface Working Group (NSI-WG)
 - Co-chairs:

Tomohiro Kudoh	t.kudoh@aist.go.jp
Guy Roberts	guy.roberts@dante.org.uk
Inder Monga	imonga@nortel.com

Dynamic Provisioning IntraDomain

Dynamic Provisioning InterDomain

- No difference from a client (user) perspective for InterDomain vs IntraDomain

IDC - Web Service Based Definition

- Four Primary Web Services Areas:
 - Topology Exchange, Resource Scheduling, Signaling, User Request

Other AAA Models Possible

- Meta-Scheduler Approach
- Same set of Web Services used for linear instantiation model can be used by a high level process to build services:
 - Topology Exchange, Resource Scheduling, Signaling, User Request
- A key issue is that this requires a trust relationship between the “meta-scheduler” and all the domains with which it needs to talk

Global Dynamic Circuit Network

InterDomain Controller (IDC) Protocol (IDCP)

- The following organizations have implemented/deployed systems which are compatible with this IDCP
 - Internet2 Dynamic Circuit Network (DCN)
 - ESNet Science Data Network (SDN)
 - GÉANT2 AutoBahn System
 - Nortel (via a wrapper on top of their commercial DRAC System)
 - Surfnet (via use of above Nortel solution)
 - LHCNet (use of I2 DCN Software Suite)
 - Nysernet (use of I2 DCN Software Suite)
 - LEARN (use of I2 DCN Software Suite)
 - LONI (use of I2 DCN Software Suite)
 - Northrop Grumman (use of I2 DCN Software Suite)
 - University of Amsterdam (use of I2 DCN Software Suite)
 - DRAGON Network
- The following "higher level service applications" have adapted their existing systems to communicate via the user request side of the IDCP:
 - LambdaStation (FermiLab)
 - TeraPaths (Brookhaven)
 - Phoebus

Open Source DCN Software Suite

- OSCARS (IDC)
 - Open source project maintained by ESNet and Internet2
- DRAGON (DC)
 - NSF-funded
 - Open source project maintained by USC ISI EAST and MAX
- Version 0.3.1 of DCNSS current deployed release
 - <https://wiki.internet2.edu/confluence/display/DCNSS>

Architecture Definition

Key Control Plane Features

- Routing
 - distribution of "data" between networks. The data that needs to be distributed includes reachability information, resource usages, etc
- Path computation
 - the processing of information received via routing data to determine how to provision an end-to-end path.
- Signaling
 - the exchange of messages to instantiate specific provisioning requests based upon the above routing and path computation functions.
- Architecture definition document under development
 - Addresses tradeoffs and decisions with respect to these issues and others.

Architecture Definition

Key Control Plane Features

- Routing
 - Topology Exchange, Domain Abstraction
 - Link State based with varying levels of dynamic information (practically no dynamic information shared in initial implementations/deployments)
- Path Computation
 - Multi-Domain, multi-stage path computation techniques (includes a Scheduling and AAA components)
- Signaling
 - path setup, service instantiation
- Architecture would allow for web service or traditional protocol types of exchanges
 - web service mechanisms are the current choice
 - future may see a mix of both

Key Control Plane Key Capabilities

IntraDomain

- InterDomain Architecture Definition and Agreements are the most important issue to be resolved amongst external organizations.
- However, there are many important IntraDomain issues as well
 - multi-layer, multi-vendor, multi-technology path computation control and provisioning
 - use of control plane protocols, management systems, or a combination of both
 - path computation and resource management which includes AAA and scheduling information
 - developing abstract view of your network for sharing with external domains
 - use of hierarchical techniques. Provision a circuit at one layer, then treat it as a resource at another layer. (i.e., Forward Adjacency concept)
- These are individual domain design decisions, but best practices and architectures will emerge

Circuit Provisioning or Network Virtualization?

OSCARS Project

- On-demand Secure Circuits and Advance Reservation System (OSCARS)
- DOE Office of Science and ESnet project
- Co-development with Internet2
- Web Service based provisioning infrastructure, which includes scheduling, AAA architecture using X.509 certificates
- Extended to include the DICE IDCP
- <http://www.es.net/oscars/index.html>

DOE

Office of
Science

DRAGON Project

- Dynamic Resource Allocation via GMPLS Optical Networks (DRAGON)
- Developed control plane for multi-technology hybrid networks
- Deployed on Internet2 HOPI and DCN
- NSF Funded Project
 - originally funded by CISE/ANIR
 - Program Manager, Kevin Thompson, OCI
- Collaborative project: USC/ISI, UMD/MAX, GMU
- <http://dragon.east.isi.edu>

Hybrid Multi-Layer Network Control Project (Hybrid-MLN)

- Investigating issues associated with Multi-Layer, Multi-Domain Hybrid Networks from an architecture, data plane, and control plane perspective
 - Architecture Development
 - Design, analysis, modeling, simulation
 - Experimentation and data collection
- USC/ISI, UNM, ESNet, ORNL, Internet2
- Funded by DOE Office of Science
 - Dr. Thomas D. Ndousse, Program Manager
- <http://hybrid.east.isi.edu>

OAK RIDGE NATIONAL LABORATORY
U. S. DEPARTMENT OF ENERGY

Commercial multi-layer network activities

- Standards bodies, vendors, commercial network deployments are working on multi-layer networks
- How do the R&E activities relate to Commercial multi-layer activities?

Standards Bodies Progress and Status

- Several standards bodies working in this space:
 - ITU-T ASON
 - OIF
 - IETF (CCAMP, L1VPN, L2VPN)
 - MEF
- UNI, I-NNI, E-NNI, BGP Extensions are some the key topics of discussion in these groups
- Vendor implementations are also following
- The work that the R&E community needs to leverage this work, but it needs to go further (and faster) then what is occurring in these standards bodies and associated vendor implementations. In particular, the standards bodies:
 - Have not converged on Inter-AS interdomain E-NNI routing or signaling protocols
 - Not working on multi-layer path computation details
 - Completed very little work on application of an Authentication, Authorization, Accounting (AAA) model to the control plane
 - Completed very little work on scheduling of provisioned services
 - Not addressing scalability and security to the degree required for the R&E community
- This is an opportunity for the R&E community to lead via early research, design, deployment of advanced multi-layer, multi-domain networks which provide real benefits to real users

Current Overview

Multi-Layer / Multi-Domain Activities

R&E Network Research and Development Moving Forward

Emerging Network Architectures

- I believe we are in the middle of a major evolution/transformation/revolution of network architectures
- This revolves around viewing the network as “multi-layer, multi-technology” construct
 - with different switching types like PSC, L2SC, TDM, LSC, FSC,
 - reflected in new technologies ethernet PBB-TE, NG-SONET, OTN, NG WDM

Emerging Network Architectures

Some Key Questions

- how do we use these layers to build the IP routed network?
- how do we use these layers to traffic engineer the IP routed network? with some humans in the loop and a couple of days of planning? in real time by “management” computers? or by the network itself?
- how do we use these layers to build totally new “network services”

Emerging Network Architectures

Some Key Questions

- the first two applications are most obvious and will provide immediate benefits
- the last item is where a new network paradigm is waiting to be developed
- Service Oriented Networks
 - networks go beyond just providing IP routed service to also providing lower layer services directly to “applications”
 - networks provide “value added service embedded in the network itself” such as high performance data backup, content distribution systems, providing virtual network topologies with flexible levels of isolation, deterministic performance, dedicated resources, user perceived performance.

Emerging Network Architectures

Some Key Questions

- Network operators of the future may need to use the power of their networks to provide value added services that can only be constructed by tight coupling with network infrastructure and capabilities.
- So the future may not be so much “network convergence” as it will be “network services convergence”
- so we will find a way to use a very heterogeneous set of network technologies to provide a rich set of network services

R&E Community Network Research

- The R&E community is uniquely positioned to make contributions in the development of future network architectures
- Commercial networks are built to support the applications (profiles) they want their customers will use
 - hope the users do not disturb the network in unexpected ways with unexpected applications
 - they have a much bigger scale dimension to address

R&E Community Network Research

- R&E community is building networks to encourage/enable domain experts to develop the applications they can imagine.
 - design networks to allow application communities to innovate, and subsequently drive the network design and requirements
- R&E networks should really view as their mission to enable/encourage application domain experts to be able to create and innovate in their domain space.

R&E Community Network Research

- This is difficult, because innovating and creating is always hard, and it is not realistic to just ask the application domain experts to tell us what they want from a next generation network. They are not network experts
- The network community needs to take responsibility for integrating deep enough into the domain areas to allow innovation to happen, and build networks to support that innovation

Summary

- Future Network architectures are likely to be based on exploiting the multi-layer topology of networks
- Networks should evolve to providing services
- Exploit and encourage the natural feedback loop between vendors, standards bodies, commercial users, commercial networks, R&E networks, R&E user.
- The future may not be so much “network convergence” as it will be “network services convergence”

Thank You

- Questions, Comments?
- Tom Lehman
 - tlehman at east.isi.edu

Extra Slides

IntraDomain Network Control

- A key requirement for the architecture is to be able to handle the reality that the underlying networks will be very heterogeneous in terms of technology, control mechanisms, and vendors.
- In the current architecture this is abstracted out by the DC to IDC interface.
- Four types of underlying domain types have been identified in terms of how the DC interacts with them:
 - GMPLS (I2 DCN is an example, regional networks based on ethernet switch dynamic provisioning is another example)
 - MPLS (ESNet SDN is an example)
 - Management Plane Controlled (USN is an example)
 - Vendor Control Plane (I2 DCN also has a component of this)

Dealing with Heterogeneous Network Technologies and Vendor Equipment

- As an Example, DRAGON is used as the DOMAIN Controller for I2 DCN Ciena Core Directors

- Adding regions of new technologies and vendors is not too difficult from the provisioning perspective
- The difficult issue is in terms of the routing exchange between/from the technology/vendor regions and path computation (intra and multi-domain) with multiple constraints.

Multi-Constraint Path Computation

- IntraDomain provisioning requires a path computation process to determine a path across the local network
- If the domain consists of multiple technologies, multiple levels, and multiple vendors this problem can be complex
- In order to realize the advanced control plane features multi-domain path computation needs to be augmented to operate in these environments. This will likely include addition of the following constraints to the path computation process:
 - time domain
 - flexible set of AAA and other user defined constraints
 - Ability to look for paths as a group in the context of a entire topology build.
 - These scheduling and flexible policy processing mechanisms will need to be tightly integrated/coupled with path computation and selection processes

Flexible and Policy Based Multiple Constraint Path Computation with Filtering/Pruning Processes

Path Computation with Multiple Dimensions

- Resource dimension
 - Link availability, bandwidth capability & resource interdependence
 - TE constraints, e.g. switching cap.
 - AAA policy dimension
 - User privileges
 - App. specific requirements (SLA)
 - Administration policies
 - Time schedule dimension
-
- Integrate and translate network resource states and policies into shared control plane intelligence.
 - Synergize AAA policy decision with TE based provisioning decision, resulting in fast, precise and simplified control process.

Optical Transport Network (OTN)

- ITU Standard G.709, Optical Transport Network (OTN), or digital wrapper technology.
- OTU1, 2.7 Gbit/s, transport a SONET OC-48 SDH STM-16 signal.
- OTU2, 10.7 Gbit/s, transport an OC-192, STM-64 or 10Gbit/s WAN. Overclocked to carry signals faster than STM64/OC192 (9.953Gbit/s) like 10 GiGE LAN PHY
- OTU3, 43Gbit/s, transport an OC-768 or STM256 signal

Provide BackBone Bridging Traffic Engineering (PBB-TE)

- 802.1q, VLANS
- 802.1ad, Provider Bridges, QinQ
- 802.1ah, PBB
 - connection oriented operation enabled by disabling flooding, learning, spanning tree
 - use a control plane to establish paths thru network
 - 24 bit service ID, eliminates scaling issue within the PBB domain
 - only switches at the edge of the PBB network need this capability, rest can be provider bridges

Internet Engineering Taskforce

CCAMP working group (GMPLS)

- GMPLS control for SONET/SDH (RFC 4257)
- GFP/LCAS interface discovery (OSPF-TE, RSVP-TE implications)
- Multi-layer/multi-region (MRN) networks drafts:
 - Interface switching capability (ISC), unified TE database
- Drafts on multi-domain routing (OSPF-TE, O-BGP), no temporal state
- Other drafts on multi-domain/AS signaling & recovery:
 - Crankback, inter-AS exclude routes, etc

Path computation element (PCE) working group (TE)

- Path composition for TE-LSP paths:
 - Centralized / distributed, loose-domain / hop-by-hop
- Inter-area / AS / layer considerations (virtual topology management)
- New PCEP signaling protocol, possibly one for PCE discovery
- No PCE considerations for advance scheduling
- Various requirements drafts (2004-5), no RFC yet

IETF L1 VPN Framework

Layer 1 VPN working group

- “Infrastructure virtualization”: DWDM lighpath, SONET circuit
- Basic and enhanced modes: signaling only vs. dist^d signaling & routing
- Drafts on BGP & OSPF PE discovery (opaque LSA), single AS focus for now
- Proposal to extend RSVP-TE signaling (per VPN instances)
- provide layer-1 VPN services (establishment of layer-1 connections between CE devices) over a GMPLS-enabled transport service-provider network.

IETF L1 VPN Service Models

Differing Levels of CE-PE Functionality / Exchange

Optical Internetworking Forum

User Network Interface (UNI) 2.0

- Multi-vendor interoperable client provisioning
 - Automated end-pt & service discovery, signaling (parameters)
- Improved resiliency, control security, Eth support (IETF, ITU-T inputs)
- UNI-N side supports multi-layer call/connections (VCAT)

Network to Node Interface (Internal – NNI, External - NNI)

- Decouple intra & inter-domain mechanisms (protocols, algorithms)
- Signaling protocol: parameter negotiation, protection/diversity
- Hierarchical routing: topology / resource discovery (DDRP mixed review)
- Generally lacks provisions for advance scheduling

IEC Supercomm interoperability trials

- Interim UNI 1.0 (2001): End-pt discovery, setup/teardown, full λ rates
- UNI 2.0, E-NNI 1.0 (2005):
 - 13 vendors, 7 service providers (focus on EoS services)

International Telecom Union (ITU-T)

Automatically-Switched Optical Network (SG - 15, G.8080)

- Multi-level hierarchical link-state routing (G.7715.x):
Horizontal (areas), vertical (leaders), inter-level state exchange
- Dist^d call / connection management (G.7713.x, SN controllers):
Recently addressing protection/restoration, no crankback yet

Layer 1 VPN (SG - 13)

- Req & architecture documents (Y.1312 / 2003, Y.1313 / 2004)
- Close liason w. IETF (routing area) on suitability of IETF protocols

Other liason activities to evolve “ASON compliant” protocols

- Signaling:
IETF RSVP-TE drafts for ASON, OIF UNI 2.0 & NNI 1.0 alignment
- Link-state routing:
 - Reqs RFC 4258, OSPF-TE and IS-IS drafts for ASON (G.7715.1)
 - OIF NNI 1.0 routing

Metro Ethernet Forum

Accelerate “carrier-class” Ethernet

- **Service focus**, layered network decomposition:
 - Applications, Eth services, metro Eth network (MEN)
- Agnostic to MEN technology (SONET, DWDM, MPLS)
- UNI spec for client-MEN boundary (UNI-C, UNI-N), NNI

E-Line Service, MEF 4 (2004)

- Point-to-point (unicast) Ethernet VC (EVC)
- Service attributes (at UNI):
 - Interfaces, BW profiles, service performance, frame delivery, service multiplexing, L2 control tunneling/discard, etc
- UNI multiplexing (EVPL service)

E-LAN Service, MEF 4 (2004)

- Multipoint-to-multipoint (broadcast) EVC
 - Best-effort or QoS between UNI's
- Similar service attributes
- Support address learning over UNI

Multi-Layer, Multi-Domain Control Plane Hybrid Networks Architecture

Current Status and Future Issues

ESCC/Internet2 Joint Techs Winter Meeting

January 22, 2008

University of Hawaii

Honolulu, Hawaii

Tom Lehman, Xi Yang
Information Sciences Institute East
University of Southern California

Chin Guok
Network Engineering Services Group
ESnet

Andy Lake, John Vollbrecht
Internet2

Nasir Ghani
University of New Mexico

Nagi Rao
Oak Ridge National Laboratory

Multi-Domain, Multi-Layer Hybrid Networks

- Hybrid networks are intended to provide a flexible mix of IP routed service and dedicated capacity “circuits”
- The “**Multi-Layer**” is meant to identify several items regarding how hybrid networks may be built. In this context it includes the following:
 - **Multi-Technology** - MPLS, Ethernet, Ethernet PBB-TE, SONET, NG-SONET, T-MPLS, WDM
 - **Multi-Level** - domains or network regions may operate in different routing areas/regions, and maybe be presented in an abstracted manner across area/region boundaries
- **Multi-Domain** indicates that we want to allow hybrid network service instantiation across multiple domains
- And of course all this implies that this will be a **Multi-Vendor** environment.

Multi-Layer, Multi-Domain Control Plane Hybrid Networks Control Plane

- Where are today?
- What is interesting to think about for the future?

Hybrid Networks Control Plane - Today

- Currently an early deployment of a Hybrid Network Control plane on ESnet SDN and Internet2 DCN
 - and some regional's such as DRAGON, NYSERNet, others
- This evolved out of collaborations between several projects including ESNet OSCARS, I2 BRUW, NSF DRAGON, and the DICE Group
- It is expected that this will evolve as standards bodies and other groups work on developing interface specifications/agreements with the larger community
- Key features of the current architecture are noted in the following bullets.
- One InterDomain Controller (IDC) per domain which is responsible for inter-domain messaging
- A “Domain Controller” (DC) which takes care of provisioning inside the domain.
 - The DC is really an internal domain concern
 - The DC design will vary by domain (based on technology types, vendor capabilities, etc), and in some instances may be a very minimal set of functions
- The IDC/DC combination provides the basis for a two-level hierarchical network view. Where the DC will have knowledge of the real local topology and the IDC may have a reduced or abstracted view.

Hybrid Networks Control Plane - Today

- Four distinct phases are identified for IDC communications: User Request, Topology Exchange, Resource Scheduling, Signaling
- Topology Exchange: currently based on abstracted link states, with little to no dynamic information.
 - We are also planning to investigate use of a path vector style of inter-domain information exchange.
- Resource Scheduling: multi-domain, multi-stage path computation process where the specific resources get identified and reserved for a specific signaling event.
- The Signaling Phase is where specific network elements are provisioned. This phase may be initiated by the user, or by the domains. The Signaling Phase actions are based on resources identified in the Resource Scheduling phase.
- User Request Phase provides a message set for users to request multi-domain circuits
- Current security and authentication models are based on signed soap messages and X509 Certificates (User to local IDC messaging; IDC to IDC messaging)

Hybrid Networks Control Plane - Today

- Four Primary Web Services Areas:
 - Topology Exchange, Resource Scheduling, Signaling, User Request

Hybrid Networks Control Plane - Today

- Meta-Scheduler Approach
- Same set of Web Services used for linear instantiation model can be used by a high level process to build services:
 - Topology Exchange, Resource Scheduling, Signaling, User Request
- A key issue is that this requires a trust relationship between the “meta-scheduler” and all the domains with which it needs to talk

Hybrid Network Control Plane

What can we do today?

- Dynamic provision of end to end (circuits) across multiple domains.
- Specify a few basic parameters regarding a single circuit request: edge technology/configuration, bandwidth, end points, domain sequence, specific start/stop times

Hybrid Network Control Plane

What is interesting to think about for the future?

- Enhanced Circuit Parameters and User Request Mechanisms
 - Richer set of flexible circuit request constraints such as technology type, flexible time period specifications, latency, jitter, adjustments to in-service circuits, arbitrary business /administrative/security constraints, flexible user requests mechanisms.
- Topology Building: combining multiple individual circuits together into a user specified topology.
- **Network Virtualization** - True Multi-Level, Multi-Technology, Multi-Vendor network control and provisioning
- Only talking about network resources here; correlation with application domain resources is considered a separate activity.

Enhanced User Options

- User has increased number of parameters to specify such as technology type, adjustments to in-service circuits, flexible time period specifications, latency, jitter, arbitrary business /administrative/security constraints, flexible user requests mechanisms

- Green Path might be chosen in response to user specifying domain preference, latency/jitter requirement, or something else

Multi-Level, Multi-Technology, Multi-Vendor Infrastructures

- Multiple Options, most will have vendor proprietary control and management mechanisms which only work across single vendor regions

Multi-Level, Multi-Technology, Multi-Vendor Infrastructures

- Current dynamic provisioning environment can be described as:

- **Static Topology, Dynamic Provisioning**

- Next we want to enable:

- **Dynamic Topology, Dynamic Provisioning**

Multi-Level, Multi-Technology, Multi-Vendor – Network Virtualization

- Network Virtualization and Topology Building in Multi-Level, Multi-Technology, Multi-Vendor Infrastructures

What are the major issues looking forward?

- A key requirement for the architecture is to be able to handle the reality that the underlying networks will be very heterogeneous in terms of technology, control mechanisms, and vendors.
- In the current architecture this is abstracted out by the DC to IDC interface.
- Four types of underlying domain types have been identified in terms of how the DC interacts with them:
 - GMPLS (I2 DCN is an example, regional networks based on ethernet switch dynamic provisioning is another example)
 - MPLS (ESNet SDN is an example)
 - Management Plane Controlled (USN is an example)
 - Vendor Control Plane (I2 DCN also has a component of this)

Dealing with Heterogeneous Network Technologies and Vendor Equipment

- As an Example, DRAGON is used as the DOMAIN Controller for I2 DCN Ciena Core Directors

- Adding regions of new technologies and vendors is not too difficult from the provisioning perspective
- The difficult issue is in terms of the routing exchange between/from the technology/vendor regions and path computation (intra and multi-domain) with multiple constraints.

Multi-Constraint Path Computation

- IntraDomain provisioning requires a path computation process to determine a path across the local network
- If the domain consists of multiple technologies, multiple levels, and multiple vendors this problem can be complex
- In order to realize the advanced control plane features multi-domain path computation needs to be augmented to operate in these environments. This will likely include addition of the following constraints to the path computation process:
 - time domain
 - flexible set of AAA and other user defined constraints
 - Ability to look for paths as a group in the context of a entire topology build.
 - These scheduling and flexible policy processing mechanisms will need to be tightly integrated/coupled with path computation and selection processes

Flexible and Policy Based Multiple Constraint Path Computation with Filtering/Pruning Processes

Path Computation with Multiple Dimensions

- Resource dimension
 - Link availability, bandwidth capability & resource interdependence
 - TE constraints, e.g. switching cap.
 - AAA policy dimension
 - User privileges
 - App. specific requirements (SLA)
 - Administration policies
 - Time schedule dimension
-
- Integrate and translate network resource states and policies into shared control plane intelligence.
 - Synergize AAA policy decision with TE based provisioning decision, resulting in fast, precise and simplified control process.

Control Plane Scalability and Performance – Simulations and Testing

- In order to collect some more data on the (current and future) control plane performance, we are planning to run some simulations on OPNET and/or User Mode Linux (UML) environments.
- This will allow us to evaluate the scalability/stability of inter-domain information exchange, the success/blocking probability/performance of Resource Scheduling and Signaling.
- User Mode Linux (UML) based simulation will also allow us to connect simulated networks to real networks, since the UML will run the same software as current networks.
- Current and future designs will be evaluated

Questions/Comments?

- Tom Lehman
 - tlehman at east.isi.edu
- Thank You

Hybrid Network Control Plane Interoperation Between Internet2 and ESnet

ESCC/Internet2 Joint Techs Summer Meeting

July 16, 2007

Fermi Lab

Batavia, Illinois

Tom Lehman

Information Sciences Institute East, University of Southern California

Chin Guok

Network Engineering Services Group, ESnet

Andy Lake, John Vollbrecht

University Corporation for Advanced Internet Development, Internet2

Hybrid Networks

Heterogeneous By Nature

- Hybrid networks are extremely heterogeneous at several levels
- DataPlane can be constructed from
 - router based Multiprotocol Label Switching (MPLS) tunnels
 - Ethernet VLAN based Circuits
 - Synchronous Optical Network / Synchronous Digital Hierarchy (SONET/SDH) circuits
 - Wavelength Division Multiplexing (WDM) connections
 - Combinations of the above

Hybrid Networks

Heterogeneous By Nature

- Control Planes can be based on
 - Multiprotocol Label Switching (MPLS)
 - Generalized Multiprotocol Label Switching (GMPLS)
 - Web Services
 - Management Systems
 - Combinations of the above
- Client (user) services or attachment points could be
 - Ethernet
 - SONET
 - IP Router
 - InfiniBand

Hybrid Networks

Web Service Control Plane Interfaces

- Web Services provides a mechanism to deal with heterogeneous control planes
 - inspired by the standards bodies work on control plane protocols, but not just recreating that work at the web service level
 - Better described as using control plane techniques to develop a “service plane”

Hybrid Networks

Web Service Control Plane Interfaces

- Four Primary Web Services Areas:
 - Topology Exchange, Resource Scheduling, Signaling, User Request

Hybrid Networks

Control Plane Architecture

- The benefits offered by Web Services include
 - standardized mechanisms for user authentication and policy management
 - flexible features for interfacing with a diverse set of I-NNI mechanisms
 - Allows focus on several issues that current control plane work has not addressed in a robust manner:
 - scalability, stability, security, flexible application of policy, AAA, scheduling
- Will still allow for peering domains with compatible non web service E-NNI (i.e. GMPLS based) to utilize that as desired
 - a domain might peer with one domain at GMPLS level, and another at the Web Service level

Web Service based E-NNI

Three Main Components

- Routing
 - Topology Exchange
 - Domain Abstraction
 - Varying levels of dynamic information
- Resource Scheduling
 - Multi-Domain path computation techniques
 - Resource identification, reservation, confirmation
- Signaling
 - path setup, service instantiation

Web Service Based Multi-Domain Provisioning – Collaborative Effort

- Architecture and Web Services Design effort is a collaborative effort between:
 - Internet2
 - ESnet
 - DANTE (GEANT, Autobahn)
- Also collaborating with University of Amsterdam on Token based RSVP signaling and topology exchange

Internet2 and ESnet

Implementation Goals and Methods

- Goal
 - Dynamically provision layer 2 circuits between Internet2 network and ESnet/SDN
- Methods
 - Use existing control plane software where appropriate
 - Provide multidomain AAA architecture for message exchanges
 - Exchange topology information
 - Coordinate the scheduling of resources
 - Signal circuit setup between domains

Control Plane Software

- OSCARS (Web Service)
 - Started by ESnet, merged with Internet2's BRUW project in 2006
 - Web service architecture, interfaces to lower level network specific provisioning systems
 - Vendor based MPLS L2VPN (Martini Draft)
- Internet2 DCS/HOPI
 - DRAGON (NSF funded project in development by USC/ISI EAST and MAX)
 - Uses GMPLS protocols to build layer 2 circuits

OSCARS Architecture

OSCARS

- Web Services
 - Source host, destination host, bandwidth, start time, end time
- “Book-ahead” scheduling
- Multidomain AAA architecture using X.509 certificates
- Uses traceroute for path calculation
- Each domain configures MPLS tunnels on Juniper routers at requested time

DRAGON

- Virtual Label Switched Router(VLSR)
 - PC based control plane software
 - Manages and provisions various network equipment such as ethernet switches, SDH/SONET
 - Signaling with RSVP packets
- Network Aware Resource Broker (NARB)
 - Stores topology in OSPF-TE database
 - Performs inter/intradomain path calculation
 - Exchanges interdomain topology

OSCARS-DRAGON Integration

AAA and Security

- OSCARS AAA
- SSL Encryption
- Authentication
 - X.509 Certificates
 - User to Domain
 - Domain to Domain
 - Web Service Security by OASIS
 - SAML assertions about end-user (future)
- Authorization
 - OSCARS attribute based system

Topology Exchange

- XML topology schema
 - Developed by DICE
 - Domain, Node, Port, Link hierarchy
 - Namespace extension to the NMWG/PerfSonar Topology Schema
- Interdomain exchanges between OSCARS servers via web services
- Topology storage flexible
 - NARB, XMDR, static file
- Path Computation by NARB or other component that uses topology data

Resource Scheduling

- OSCARS web service message
 - Source port, destination port, bandwidth, start time, end time
 - Optional parameters such as VLAN tag
- Generates information important to later signaling
 - Global-ID, secure token
- Stored in OSCARS reservation database

Signaling

- XML signaling
 - OSCARS receives XML signaling message
 - Contacts VLSR to initiate path setup
 - Could also talk to Juniper router or other device
- RSVP signaling
 - User signals VLSR directly
 - Token-based signaling developed by University of Amsterdam's Phosphorous project
 - Token, Global Reservation ID

Timelines and Future Work

- Timeline
 - Hope to have test release by end of Summer 2007
 - Official release in Fall 2007
- Future Work
 - Continue collaboration with international partners
 - Explore use of federated identity software such as Shibboleth

Questions/Comments?

- Tom Lehman (tlehman@east.isi.edu)
- Chin Guok (chin@es.net)
- Andy Lake (alake@internet2.edu)
- John Vollbrecht (jrv@internet2.edu)

Data Plane Testing on USN, ESnet, and Internet2 Networks

ESCC/Internet2 Joint Techs Summer Meeting

July 16, 2007

Fermi Lab

Batavia, Illinois

**US Dept. of Energy
Office of Science**

Tom Lehman (USC/ISI)
Nasir Ghani (Tennessee Tech)
Chin Guok (ESnet)
Nagi Rao (ORNL)
John Vollbrecht (Internet2)
John Moore (MCNC)

Hybrid MLN

OAK RIDGE NATIONAL LABORATORY
Managed by UT Battelle for the Department of Energy

Outline

- Dataplane testing configuration
- Dataplane test data
- Dataplane simulation
- Dataplane test conclusions
- Future Work

Hybrid Network Data Planes

- Basic premise of hybrid networks is the availability of both best effort routed service and deterministic dedicated resource paths, i.e., circuits
- There are many technologies available over which to construct these circuits
 - IP router-based Multiprotocol Label Switching (MPLS) Label Switched Paths (LSPs) “circuits”
 - Ethernet VLAN based “circuits”
 - SONET/SDH TDM “circuits”
 - Wavelength Division Multiplexing (WDM) “circuits”

What Data Plane Technologies to Use?

- What do you want to do with your circuits?
 - Dedicated bandwidth connections for deterministic file transfers?
 - Dedicated bandwidth & low jitter for instrument control or interactive applications?
 - Connector backhaul to your IP Network?
 - Traffic engineering of your IP Network?
 - Dynamic router-to-router circuits for traffic cut thru?
 - Computer to Computer communications?
 - Processor to memory? block data storage system access?
 - Setting up application specific topologies to create & optimize distributed application or data storage systems?

Data Plane Testing

- Test characteristics/performances of “circuits” constructed via different technologies; and also “end-to-end paths” constructed via concatenation of individual circuits
- Questions
 - What is difference between the different technologies?
 - How well does the concatenation/stitching work?
 - How well does policing/shaping work at the edge?
 - What happens to a flow that is policed/shaped at the ingress edge by the time it exits the egress edge?

Data Plane Testing

Data Paths Across:

- ESnet
- USN
- Abilene
- Internet2 Network
- DRAGON

		Data Plane Technology			
		PSC	L2SC	TDM	LSC
Network	USN		X	X	
	SDN	X	X		X
	ESNet	X			
	I2 Abilene	X			
	I2 DCS/HOPI		X	X	X
	DRAGON	X	X		X

<http://hybrid.east.isi.edu>

Test Equipment

- Spirent AX4000 - Hardware based Traffic Source and Sink
- External CDMA Clock allows for synchronized timestamps

Spirent AX4000

- 10 Gbps with OC192 POS / BERT / 10GbE
- Two Gigabit Ethernet

Data Collection

- Approximately 75 individual tests (generally a unique path)
 - tests were sourced from 1 Gbps interfaces, some from 10 Gbps
 - some of the measured flows had cross traffic introduced
- Tests generally included 9 measured data collection runs
 - 64, 500, 8000 byte MTU
 - 100, 500, 800 Mbps for 1 Gbps paths
 - 1, 5, 8 Gbps for 10 Gbps paths
- For each test run, the following data was collected:
 - average datarate
 - total packet loss
 - average latency
 - jitter profile (histogram)
 - transfer delay (histogram)

HOPI-Abilene-UltraScience Net-ESnet Test

Histograms

packet inter-arrival

packet end-to-end delay

Spirent source → Washington-Force 10 → Washington-Juniper T640 → Chicago-Juniper T640 → Chicago-Force 10 → Chicago-Glimmer Glass → Chicago-Force 10 → Chicago-Cisco 6509 → Seattle-Juniper T640 → Sunnyvale-Juniper T640 → Sunnyvale-Force 10 → Sunnyvale-CDCI → Seattle-CDCI → Chicago-CDCI → Chicago-Force 10 → Chicago-Juniper T640 → Washington-Juniper T640 → Washington-Force 10 → Spirent receiver

Circuit Description Example

- The formal description of this extended inter-network path:
 - Circuit type:
 - *usn [ethernet:tdm:ethernet]:i2dsn [ethernet:tdm:ethernet]:esnet [ethernet:pscq:ethernet]:usn[ethernet:tdm:ethernet]*
 - Circuit path:
 - *usn [ORNL:CHIN]:i2dsn [CHIN:WASH]:esnet [WASH:CHIN]:usn [CHIN:STTL:SUNV]*

Modeling & Simulation

OPNET Modeler™ Environment

Overview

- Discrete event simulation
- GUI interface, high re-use
- Full C/C++ interface
- Hierarchical modeling:
Subnet-node-link-process

“In-House” Development

- MPLS/GMPLS control:
RSVP-TE, OSPF-TE, PCE
- Layer 2/3 data plane:
IP/MPLS, VLAN
- Full Layer 1 support:
DWDM, SONET, GFP
- Model any networks

Test Plans, Reports, and Data Repositories

- Test Report and Plans
 - <http://hybrid.east.isi.edu> → DataPlane Testing and Analysis
- Raw data repositories
 - <http://www.csm.ornl.gov/ultranet/SpirentMeasurements/>
 - <http://hpn.east.isi.edu/dataplane/sprint-test-data/>

Summary/Conclusions

- All of the tested networking technologies (PSC, L2SC, TDM, LSC) and networks (ESnet, USN, Abilene, HOPI, DRAGON) performed well both individually and when concatenated together
- There are some key differences observed between the various networking layer technologies when driven at or close to bottleneck capacity
 - QoS techniques applied to router MPLS or Ethernet switched paths exhibited notably different delay behaviors versus dedicated circuit-paths (TDM)
 - TDM-based infrastructures is most germane for applications requiring stringent guarantees on latency, jitter, and bandwidth protection
- Inter-layer cross-connections can be achieved in a reasonable manner by “stitching” together different network layer technologies.
 - Ethernet VLANs presents the least problematic demarc (automated techniques needed to coordinate VLAN tag space)
- Future Work
 - impact of ingress traffic “burstiness” on end-to-end delay and loss profiles, i.e., both for reference and interfering cross-traffic streams.
 - best techniques for ingress policing and transit node QoS
 - vendor interoperability testing
 - additional network testing

Thank-You

Questions & Comments ?

Tom Lehman
tlehman@east.isi.edu

Multi-Level, Multi-Domain Hybrid Network Inter-Operation & Performance

Joint Techs Winter Meeting

February 12, 2007

Minneapolis, Minnesota

US Dept. of Energy
Office of Science

Tom Lehman (USC/ISI)
Nasir Ghani (Tenn Tech)
Chin Guok (ESnet)
Nagi Rao (ORNL)
John Vollbrecht (Internet2)
John Moore (MCNC)

Outline

- Project Overview
- Multi-Layer, Multi-Domain Hybrid Network Issues/Investigation Areas
- Hybrid Network DataPlane(s)
- Hybrid Network Control Plane(s)

Project Overview

- Hybrid Multi-Layer Network Control Project
- Funded by DOE Office of Science
 - Dr. Thomas D. Ndousse, Program Manager
- Investigating issues associated with Multi-Layer, Multi-Domain Hybrid Networks from an architecture, data plane, and control plane perspective
 - Design and analysis
 - Experimentation and data collection
 - Modeling and simulation

Hybrid MLN Participants

- Tennessee Tech University
 - Nasir Ghani
 - Qing Liu
- Information Sciences Institute East
 - Tom Lehman
 - Xi Yang
- Internet2
 - Rick Summerhill
 - John Vollbrecht
 - Andrew Lake
- Oak Ridge National Laboratory
 - Nagi Rao
- ESnet LBNL
 - Chin Guok

Hybrid Network Data Planes

- Basic premise of hybrid networks is the availability of both best effort routed service and deterministic dedicated resource paths, i.e., circuits
- There are many technologies available over which to construct these circuits
 - IP router-based Multiprotocol Label Switching (MPLS) Label Switched Paths (LSPs) “circuits”
 - Ethernet VLAN based “circuits”
 - SONET/SDH TDM “circuits”
 - Wavelength Division Multiplexing (WDM) “circuits”

What Data Plane Technologies to Use?

- What do you want to do with your circuits?
 - Dedicated bandwidth connections for deterministic file transfers?
 - Dedicated bandwidth & low jitter for instrument control or interactive applications?
 - Connector backhaul to your IP Network?
 - Traffic engineering of your IP Network?
 - Dynamic router-to-router circuits for traffic cut thru?
 - Computer to Computer communications?
 - Processor to memory? block data storage system access?
 - Setting up application specific topologies to create & optimize distributed application or data storage systems?

Data Plane Testing

- Test characteristics/performances of “circuits” constructed via different technologies; and also “end-to-end paths” constructed via concatenation of individual circuits
- Questions
 - What is difference between the different technologies?
 - How well does the concatenation/stitching work?
 - How well does policing/shaping work at the edge?
 - What happens to a flow that is policed/shaped at the ingress edge by the time it exits the egress edge?

More Questions

- What are the performance characteristics (jitter, loss, latency) of circuits provisioned using individual data plane technologies?
- What are the performance characteristics of end-to-end circuits constructed via concatenation of multiple individual circuits (spanning multiple data plane technologies & network domains)
- What is the performance of an end-to-end circuit constructed via the hierarchical nesting of circuits at different technology levels?
- How to quantify and characterize the performance of data plane technologies so that network designers and end-users can make decisions as to what is best suited to their needs and objectives?
- Can analytical and simulation models be built to correlate empirical findings of real world data collection to a reasonably close degree?
- Can these analytical/simulation models further predict performance in generalized hybrid networks and assist w. design efforts?

Data Plane Testing

- Circuit Descriptions
 - Circuit type:
 - *network [accessframing:dataplane:accessframing]*
 - Circuit path:
 - *network [ingressloc:transitnodes:egressloc]*
- Circuit Type:
 - *network[accessframing:dataplane:accessframing]*
 - Where the following values are possible for the above parameters:
 - Network - esnet, sdn, abilene, i2dsn, hopi, usn, dragon
 - Accessframing - ethernet, sonet (not included in this test plan), infiniband (not included in this test plan)
 - Dataplane - psc, pscq, l2sc, tdm, lsc (where pscq is a PSC path with QoS applied to the LSP)

Data Plane Testing

Multi-Layer, Multi-Domain LSPs

Best Effort (IP)- Packet (PSC) - SONET (TDM) – Best Effort (IP)

Packet (PSC) – SONET (TDM) – Ethernet (L2SC)-Packet (PSC)

An Example Circuit

- The formal description of this extended inter-network path:
 - Circuit type:
 - *usn [ethernet:tdm:ethernet]:i2dsn [ethernet:tdm:ethernet]:esnet [ethernet:pscq:ethernet]:usn[ethernet:tdm:ethernet]*
 - Circuit path:
 - *usn [ORNL:CHIN]:i2dsn [CHIN:WASH]:esnet [WASH:CHIN]:usn [CHIN:STTL:SUNV]*

Data Plane Testing

- Initial Testing Configuration, using Spirent AX4000
- More testing/paths to follow after this round complete

Data Paths Across:

- ESnet
- USN
- Abilene
- Internet2 Network
- DRAGON

		Data Plane Technology			
		PSC	L2SC	TDM	LSC
Network	ESNet	X			
	USN		X	X	
	Abilene	X			
	I2Net		X	X	X
	DRAGON	X	X		X

<http://hybrid.east.isi.edu>

Test Equipment

- Spirent AX4000 - Hardware based Traffic Source and Sink
- External CDMA Clock allows for synchronized timestamps

Spirent AX4000

- 10 Gbps with OC192 POS / BERT / 10GbE
- Two Gigabit Ethernet

Modeling & Simulation

OPNET Modeler™ Environment

Overview

- Discrete event simulation
- GUI interface, high re-use
- Full C/C++ interface
- Hierarchical modeling:
Subnet-node-link-process

“In-House” Development

- MPLS/GMPLS control:
RSVP-TE, OSPF-TE, PCE
- Layer 2/3 data plane:
IP/MPLS, VLAN
- Full Layer 1 support:
DWDM, SONET, GFP
- Model any networks

Models Suite (Data Plane)

- Layer 2/3 nodes (IP/MPLS, VLAN)
 - Generic modeling of T640, E300, others
 - Advanced I/O buffering design, policing, scheduling
 - Status: MPLS completed (12/06), VLAN nearly done
- Layer 1 nodes (SONET, DWDM)
 - SONET TSI, GFP edge mappings (model Ciena CDI)
 - DWDM cross-connects w. SONET framing
 - Status: Completed (1/07)
- End-systems & test applications
 - TCP & UDP file, TCPMON, UDPMON, ICMP-PING, ext. timestamps (SPIRENT tester)
 - Status: All completed except TCPMON (12/06)

Simulation Test Plan

- Current activities
 - Coding wrap-up, active development/testing of end-to-end delay and jitter scenarios
 - Corroborating simulations with recent live tests on ESnet-USN, USN-Cheetah, USN-Hopi, etc
 - Gauge impact of SONET segments in e-2-e path
- Near term & future activities
 - Extrapolate out and simulate larger networks & domain mixtures (VLAN-SONET-MPLS)
 - Generate inputs for subsequent control plane development phase (i.e., TE routing rule-sets)

Thank-You

Questions & Comments ?

Tom Lehman
tlehman@isi.edu