Multilayer Networks: An Architecture Framework

Tom Lehman and Xi Yang, University of Southern California Information Sciences Institute
Nasir Ghani and Feng Gu, University of New Mexico
Chin Guok, Inder Monga, and Brian Tierney, Lawrence Berkeley National Lab

ABSTRACT

We present an architecture framework for the control and management of multilayer networks and associated advanced network services. This material is identified as an "architecture framework" to emphasize its role in providing guidance and structure to our subsequent detailed architecture, design, and implementation activities. Our work is motivated by requirements from the Department of Energy science application community for real-time on-demand science-domain-specific network services and resource provisioning. We also summarize the current state of deployments and use of network services based on this multilayer network architecture framework.

INTRODUCTION

Emerging paradigms for next-generation network architectures revolve around the notion of the network as a heterogeneous "multilayer, multitechnology" construct over which multiple "services" can be provided. These services include traditional IP-routed services as well as native access services from lower layers based on technologies such as multiprotocol label switching (MPLS), Ethernet, Ethernet provider backbone bridge (PBB), synchronous optical networking (SONET)/synchronous digital hierarchy (SDH), next-generation SONET/SDH, and wavelength-division multiplexing (WDM). It is envisioned that such direct access to and control of lower layers will enable advanced traffic engineering of IP routed networks, along with the provision of advanced services tailored to application-specific requirements. A critical aspect of emerging infrastructures is heterogeneity with regard to technologies, services, vendors, and other areas. In this article we focus on the following dimensions of network heterogeneity:

Multiservice: This term refers to the client experience when connecting to the edge of a network. Since there are often multiple service options, their associated service definitions can be varied based on the underlying network implementations. For example, typical service definitions are characterized by the combination of the physical port type (e.g., Ethernet, SONET/SDH, Fibre Channel), network transport instance (e.g., IP routed, Ethernet virtual LAN [VLAN], SONET), and performance characteristics (e.g., bandwidth, delay, jitter).

Multitechnology: This term refers to the possible deployment of multiple technologies to implement the required network services. For example, operators may use technologies such as IP, Ethernet, MPLS, T-MPLS, SONET, next-generation SONET, and WDM.

Multilevel: This term refers to the fact that domains or network regions may operate in different routing areas and be represented in an abstract manner across associated area/region boundaries.

Multilayer: This term describes an abstraction that encompasses both the concepts of multilevel and multitechnology as defined above.

Along these lines, in this article we present an architecture framework for the control and management of a multilayer network (MLN) and its associated advanced network services. This material is identified as an "architecture framework" to emphasize its role in providing guidance and structure to our subsequent detailed architecture, design, and implementation activities. Overall, our efforts are motivated by requirements from the U.S. Department of Energy "e-science" application community for real-time on-demand specialized network services and resource provisioning (e.g., to support bulk transfers, real-time visualization, remote steering, etc.). We also summarize the current state of deployments and the use of network services based on this MLN architecture framework.

The remainder of this article is organized as follows. First, we describe a generic architectural framework for networks. Next, we provide further discussions on several multilayer network architecture and design alternatives. We then present a concept of service workflows to demonstrate the notion of coordinated multilayer control. Finally, we present the status of current work and future plans.

 0163-6804/11/$25.00 © 2011 IEEE

CAPABILITYPLANES

CAPABILITYPLANES OVERVIEW

The primary focus of many advanced multilayer network architectures has been the data plane technology types and the various ways they may be combined. However, there are other critical capabilities and functions required in order to manage and control complex next-generation network architectures. To facilitate the discussion and definition of our MLN architecture framework, we first introduce the concept of a "capability plane," or CapabilityPlane. A CapabilityPlane represents a grouping of related functions that are needed for the operation of a single technology layer in our multilayer network construct. Our CapabilityPlane definition spans the functional areas necessary to provide client services, as user requirements are a primary motivation for our work. We first define different CapabilityPlanes for a generic technology layer. With these foundations in place, we can then examine the use of CapabilityPlanes in the construction and operation of multilayer architectures that combine two or more of the technology layers. In particular, the following CapabilityPlane types are identified: DataPlane, ControlPlane, ManagementPlane, AAPlane, ServicePlane, and ApplicationPlane. These are now detailed further.

DATAPLANE

The DataPlane is the set of network elements that send, receive, and switch 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 that are grouped together and utilize the same DataPlane *technology type*. The technology types are defined using the standard generalized MPLS (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)

Each of these technology types includes unique features and capabilities. Furthermore, it is well understood that most networks of interest will be constructed by utilizing a combination of two or more of these DataPlane technologies to form a multilayer network, as discussed later. As a result, there are multiple implementation options in terms of the technology types (IP routing [with MPLS capabilities], Ethernet, Ethernet PBB, SONET/SDH, next-generation SONET/SDH, WDM, etc.). A detailed discussion of these specific technology types is not a topic for this article. Instead, we identify the key functions for all the CapabilityPlanes. The key functions for the DataPlane are identified as follows:

Element control: This refers to the ability to send, receive, and switch network data. The specific control functions 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 adaptation: 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 that have Ethernet client ports which are adapted into SONET/SDH or WDM for transmission over wide-area links.

CONTROLPLANE

The ControlPlane is responsible for the control and provisioning functions associated with 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 that directly interact with the DataPlane. The key functions identified for this plane are as follows:

Routing: Routing protocols are responsible for the reliable advertisement of the network topology and the available resources (e.g., 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 a network region. Within a given DataPlane technology region, the routing information may be either distributed or centralized.

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 multilayer, multivendor, multidomain environment, this may require many specific discrete functions such as traffic engineering database pruning, network graph construction and transformations, multidimensional constrained shortest path first (CSPF) computations, and application of many other possible algorithms.

Signaling: Signaling protocols are responsible for provisioning, maintaining, and deleting connections, and the exchange of messages to instantiate specific provisioning requests based on the above routing and path computation functions. This may be accomplished via protocols such as Resource Reservation Protocol — Traffic Engineering (RSVP-TE) [2] or web-service-based messaging, for example.

Traffic engineering database (TEDB): This is the function inside the ControlPlane that stores the topology state of the DataPlane. The information in the TEDB will come from the routing

The ControlPlane is responsible for the control and provisioning functions associated with the DataPlane. This generally includes maintaining topology information and configuring network elements in terms of data ingress, egress, and switching operations.

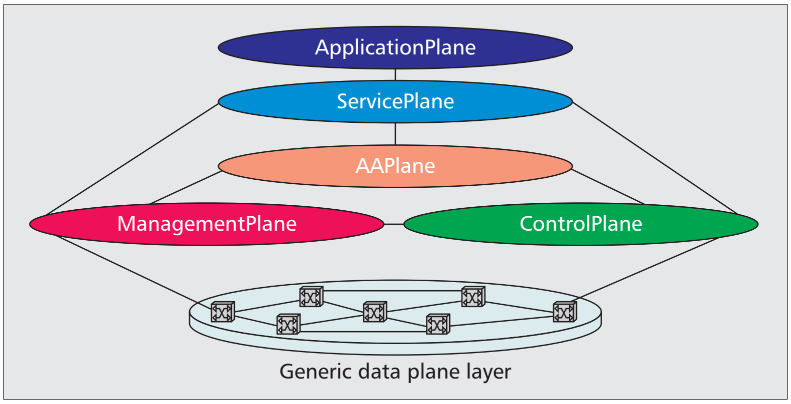

Figure 1. *CapabilityPlanes organization.*

information as well as from external sources such as the other CapabilityPlanes.

ControlPlane Status: This function involves the maintenance of ControlPlane state such that recovery mechanisms can restore operation after ControlPlane failures.

MANAGEMENTPLANE

The ManagementPlane refers to the set of systems and processes that are utilized to monitor, manage, and troubleshoot the network. The ManagementPlane is one of two CapabilityPlanes that directly interact with the DataPlane. The ManagementPlane may be queried by network administrators, users, and other CapabilityPlanes such as the ServicePlane or ApplicationPlane. The ManagementPlane may publish monitoring data, or metadata, which is available for other domains to access. The key functions identified for the ManagementPlane are as follows:

Monitoring: This function involves querying each of the other CapabilityPlanes for information. For the DataPlane this may include information such as network element status, utilization information, resource configuration, and interface information (errors, utilization, configurations). For each CapabilityPlane, status monitoring will be defined that is unique to its function set.

Troubleshooting procedures: This function involves the investigation of issues and problems in the network. These 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.

AAPLANE

The AAPlane is the authentication and authorization plane, and is responsible for the mechanisms which allow the other planes to identify and authenticate users and receive associated policy information. The key functions identified for the AAPlane are as follows:

AuthN: This is the function that identifies the user. It takes some type of user credential (e.g., a username and password, a signed certificate, or some other identity provider handle), verifies that credential, and then returns the local identifier and possibly attributes for the user.

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 defines the authorization parameters.

AAPlane Status: This function involves the maintenance of AAPlane state such that recovery mechanisms can restore operation after AAPlane failures (e.g., corruption of the authentication or authorization policies).

SERVICEPLANE

The ServicePlane refers to the set of systems and processes that are responsible for 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 and coordinate associated workflow processes. The ServicePlane also has a very important role in the coordination of other CapabilityPlanes' actions to achieve a higher-level goal. The key functions identified for the ServicePlane include processing service requests and subsequently coordinating with the other CapabilityPlanes to service the requests. This function is realized primarily in the form of CapabilityPlane service workflows, which are discussed later. Overall, the key functions identified for the ServicePlane are as follows:

Service management: This involves processing service requests and coordinating with the other CapabilityPlanes to service the requests. For interdomain or intertechnology 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. 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 (e.g., the ability to specify latency or jitter requirements).

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 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 supports 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 Capabilty-Planes.

APPLICATIONPLANE

The ApplicationPlane provides higher-level functions that 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 that 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. The key functions identified for the ApplicationPlane are as follows:

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 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 article. 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.

CAPABILITYPLANE SUMMARY

Figure 1 depicts the concept of operation for how these CapabilityPlanes are interconnected and interact. As noted above, the only planes that actually touch the DataPlane are the ControlPlane and ManagementPlane. Another way to view the DataPlane layer to CapabilityPlane relationship is from a DataPlane layer centric view. Meanwhile, Fig. 2 depicts the network capabilities of the DataPlane in terms of layers 1, 2, and 3. In this figure the capabilities that are identified for each layer correspond to the CapabilityPlanes discussed earlier. 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 PSC on top of LSC equipment, there may be a desire to have a single CapabilityPlane which is responsible for both the PSC and LSC technology regions. This would be acceptable within the framework of this architecture description. The purpose of the multilayer architecture described herein is to define functions and capabilities. Implementation options (e.g., combining CapabiltyPlanes into a single instantiation) are indeed compatible with the concepts presented here. The remainder of this article focuses on describing the architecture from a CapabilityPlane perspective.

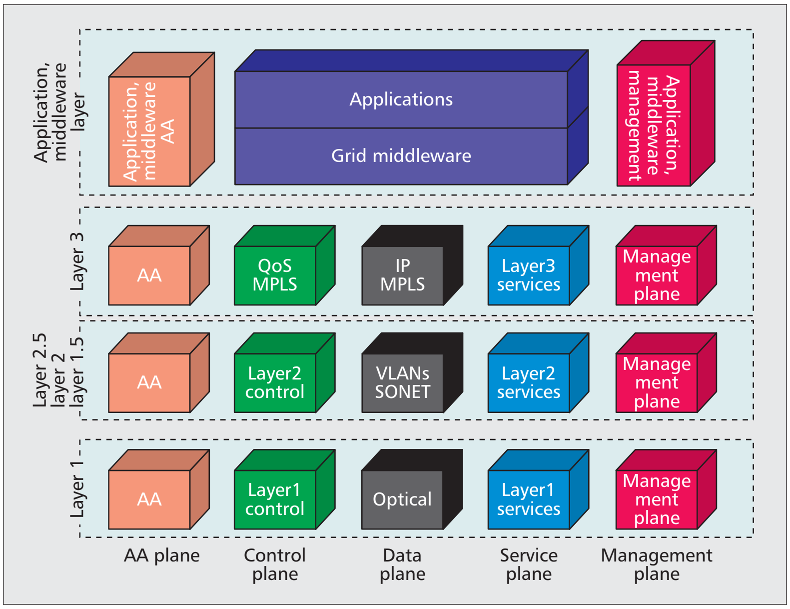

Figure 2. *Layer view of capabilities.*

MULTILAYER NETWORK ARCHITECTURES

In this section we consider the design of networks with two or more technology regions/layers. In fact, the majority of networks deployed today consist of multiple technology layers, with routers over WDM being one common example. Different technology layers are selected based on a variety of technical and practical considerations. However, the layers are generally treated as separate and distinct, and not managed together in the multilayer context we describe here. We expect that existing networks as well as new deployments will continue to be multilayer, and there will be increasing interest in integrated multilayer control and management capabilities. This is a key motivation for our work. Our descriptions of multilayer and multitechnology networking here are similar to those described in the Internet Engineering Task Force (IETF) requirements for GMPLS-based multiregion and multilayer networks (MRN/MLN) [3]. In addition to the network layer concept, which is primarily from a DataPlane topology and connection perspective, the concept of layer-associated CapabilityPlanes provides us with the added flexibility to explore multiple possible configurations for such networks. We classify them into four MLN models: MLN-Vertical, MLN-Horizontal, MLN-Combined, and MLN-InterDomain.

MULTILAYER NETWORKING — VERTICAL

First we discuss a multilayer network where two or more DataPlane types may be layered in a vertical manner. The notion of vertical layering of technology regions implies using lower-layer

We expect that existing networks as well as new deployments will continue to be multi-layer and there will be increasing interest in integrated multi-layer control and management capabilities. This is a key motivation for our work.

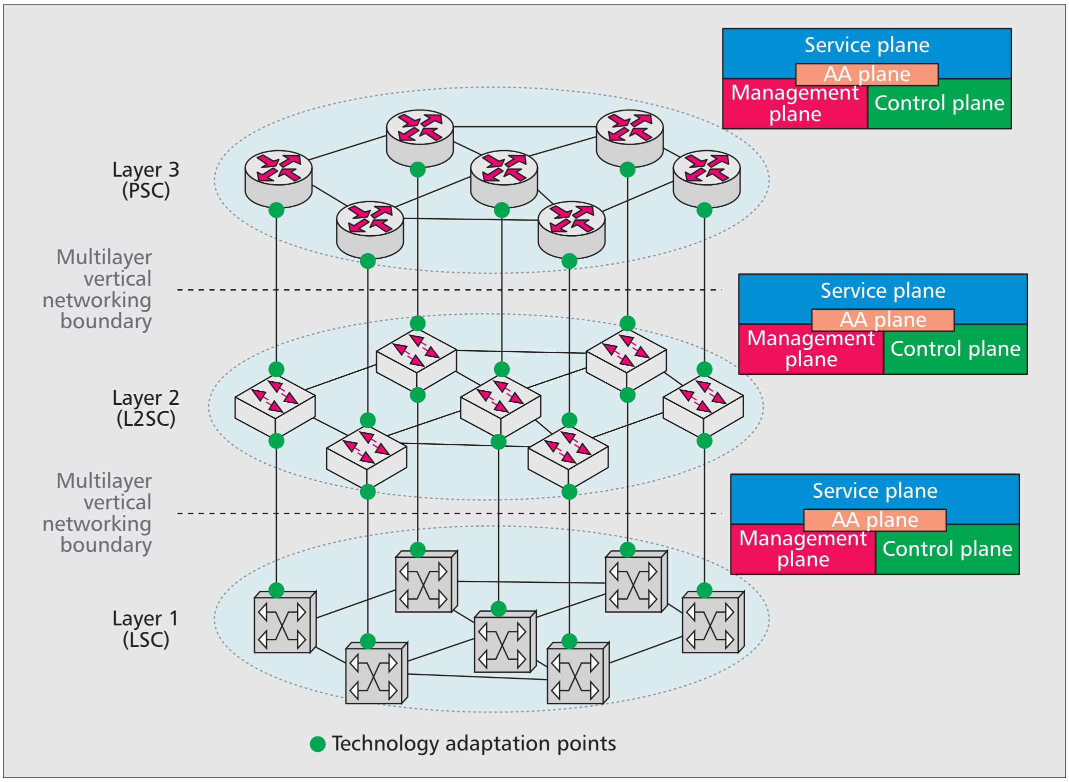

Figure 3. *Multilayer networking — vertical.*

service provisioning to provide capabilities at the higher layers. Specifically, Fig. 3 depicts a vertical multilayer 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. A typical provisioning action for a vertical multilayer topology such as this would be for a lower layer, such as the LSC region, to provision a circuit that 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.

MULTILAYER NETWORKING — HORIZONTAL

In this subsection we discuss a multilayer network topology where two or more DataPlane types may be layered in a horizontal manner. Here, Fig. 4 depicts the notion of a horizontal layering of technology regions. This topology implies that we are integrating or "stitching" services across different technology region boundaries; as opposed to 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 multilayer 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 (e.g., 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 also required in order to move data across technology region boundaries.

MULTILAYER NETWORKING — COMBINED

We can also combine the vertical and horizontal multilayer topologies to create a more flexible and sophisticated set of available network services. This is shown in Fig. 5, where a topology may consist of vertical multilayer networks peering in a horizontal manner. In this context, all the peering links (i.e., links that cross the boundary line) represent horizontal multilayer networking. Hence, there are two such links shown in Fig. 5, but there could be more.

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 it may have been created via an earlier provisioning of some resources from the horizontal lower-layer link (layer 1 in this example) . In either case, the architecture and subsequent handling of provisioning events will be identical. Note that 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 still remains.

MULTILAYER NETWORKING — INTERDOMAIN

The notion of InterDomain messaging and service provisioning is similar to the horizontal MLN case, where the horizontal boundary line is also a domain boundary. However, the primary

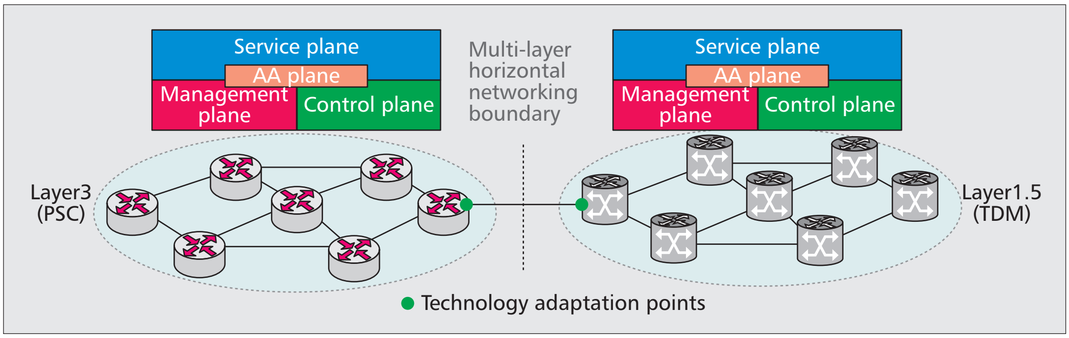

Figure 4. *Multilayer networking — horizontal.*

The most obvious scenario for the nested case is the provision of an entire topology that is handed off from one ServicePlane to a second ServicePlane (and associated CapabilityPlanes), which then assumes responsibility for subsequent service instantiations.

difference between the two is a matter of tailoring the full set of interregion communications to a subset that will meet the security and scalability requirement for InterDomain communications. This model is also illustrated in Fig. 5.

HYBRID MULTILAYER NETWORKING AND CAPABILITYPLANES

The concept of hybrid networking is introduced here in the context of using multilayer 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 article. However, this MLN topic area is deemed as one of key importance, and there is a clear need for continued research, development, and even deployment of solutions.

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

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. nonnested concept. Hence, 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 that is handed off from one ServicePlane to a second ServicePlane (and associated CapabilityPlanes), which then assumes responsibility for subsequent service instantiations.

The MLN-Vertical type is an example of a nested CapabilityPlane action. However, the nested capability plane concept is really broader than that limited example. Specifically, this topic is intended to encompass a larger range of functions and systems, which utilizes MLN capabilities in sophisticated and complex ways to create and manage multiple "virtual" network topologies. These topologies may appear independent to some CapabilityPlanes but in reality are created via a recursive handoff of resource subsets 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 article. This is another topic that requires additional research and development by the broader MLN community.

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

CAPABILITYPLANE SERVICE WORKFLOWS

A control or configuration action on a network will typically be the result of a workflow process that coordinates actions across multiple CapabilityPlanes. This is referred to as a multiple CapabilityPlane service 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. As a result, there are potentially many types of workflows, each involving all or a subset of the CapabilityPlanes. Along these lines, Fig. 6 depicts a simple workflow associated with a user requesting a circuit instantiation across two network domains. The coordinated

Overall, the current set of deployed networks only capitalize upon a small subset of the many features introduced in the proposed MLN framework. However, increasing requirements will inevitably drive broader adoption of the MLN framework.

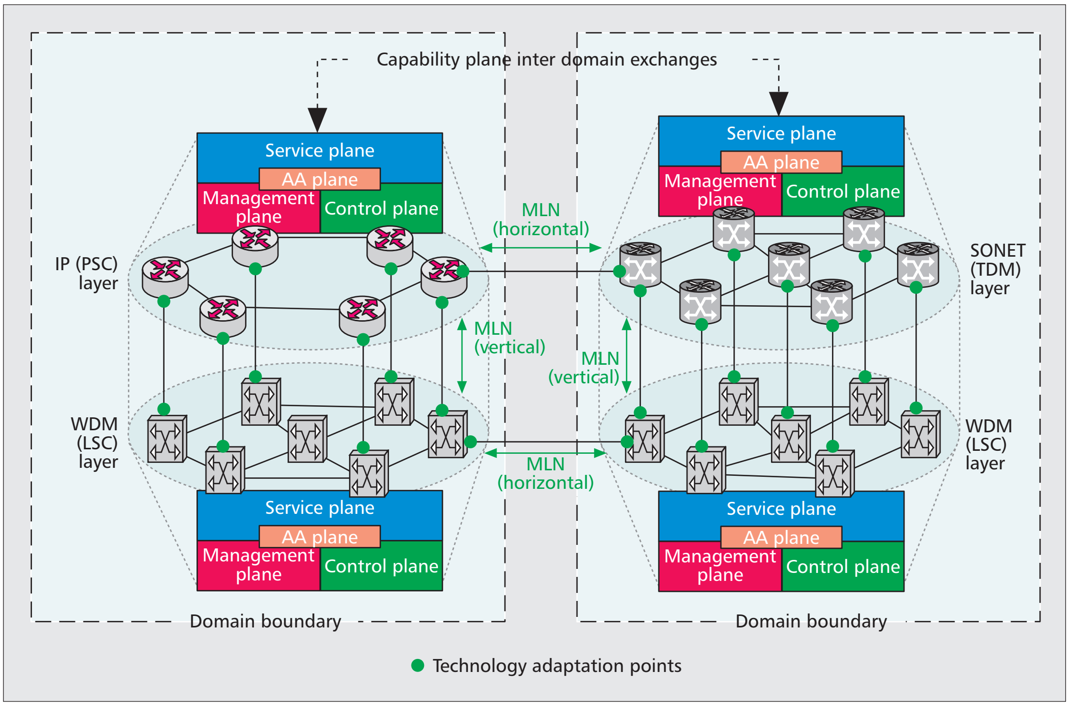

Figure 5. *Multilayer networking — combined.*

workflow includes interdomain ServicePlane interactions, which result in specific AAPlane and ControlPlane actions in their respective domains. The result is a circuit instantiation across each DataPlane that is "stitched" together to operate as a single end-to-end circuit from the user perspective. The ServicePlane is the CapabilityPlane with primary responsibility for initiating and managing these workflows in response to ApplicationPlane requests.

Deployment Status and Next Steps

Multiple networks have deployed systems implementing a subset of the MLN architecture framework described here. This includes deployments on ESnet SDN [4], Internet2 ION [5], USLHCnet [6], and others. In particular, the ESnet OSCARS [7, 8] software suite provides the CapabilityPlane implementation for these networks. OSCARS has focused on the ServicePlane and ControlPlane functionalities. These deployments are being utilized today to provide advanced network services to DOE Office of Science applications in the area of high-energy physics, climate modeling, and others.

The primary service provision at this time takes the form of dedicated bandwidth virtual circuits provisioned in real time based on user requests. The service interface technology and user demarcation is generally Ethernet VLAN. Here, the decision to select Ethernet VLAN as the service transport is twofold. First, since Ethernet technology is ubiquitous, it provides a common DataPlane framing protocol that can stitch end-to-end virtual circuits across domain boundaries (e.g., between the user and provider, and between providers). Second, by utilizing Ethernet VLANs, multiple virtual circuits can be instantiated over a single physical port. This reduces overheads and facilitates ease of deployment.

While the service interface technology has been normalized based on Ethernet VLANs, the DataPlane technology utilized to provide this service is diverse, and can include IP routers (PSC), SONET switches (TDM), and Ethernet devices (L2SC). For example, the ESnet and Internet2 DataPlanes are based on Ethernet over MPLS. Meanwhile, the USLHCNet DataPlane is based on a mix of Ethernet over SONET and native Ethernet network elements. In all cases, OSCARS leverages the vendor capabilities to provide the ServicePlane, ControlPlane, and AAPlane functional elements of the MLN architecture described. Hence, in order to create an end-to-end multidomain circuit, each domain provisions a virtual circuit that is stitched across peering points and utilizes the MLN framework to provide a uniform end-to-end Ethernet VLAN service.

Overall, the current set of deployed networks only capitalize on a small subset of the many features introduced in the proposed MLN framework. However, increasing requirements (driven by more stringent user needs, simplifying service offerings and delivery, and reducing cost and increasing efficiency) will inevitably drive broader adoption of the MLN framework. By using the MLN framework in conjunction with intelligent complex constrained path finding algorithms, networks will be able to provide a better user experience by determining the best DataPlane transport to meet the user requirements, abstract technology specific complexities of the DataPlane from the ServicePlane, and reduce cost by utilizing the lowest network layer in the DataPlane that complies with the service level

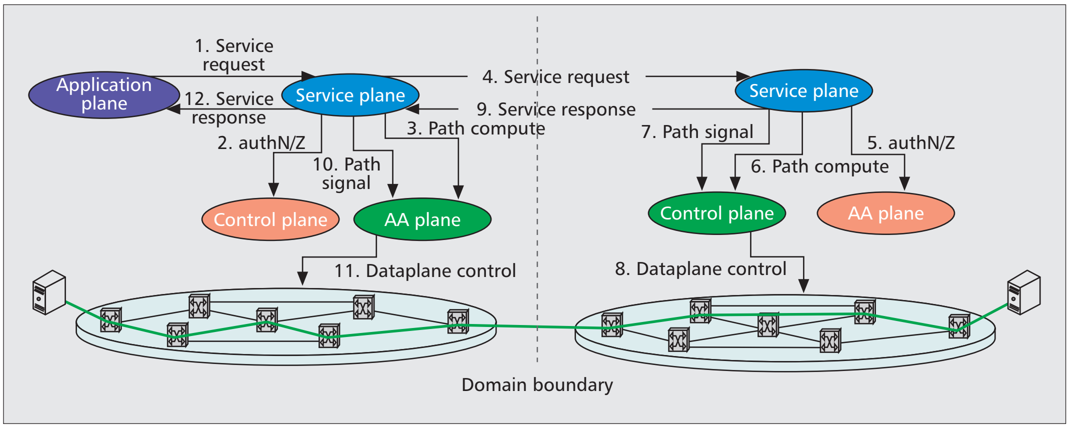

Figure 6. *Simple CapabilityPlane service workflow.*

The concepts described in this article are intended to present a vision for moving forward to realize the seamless and dynamic movement of data flows, services, and virtualized network environments across all layers of network infrastructures.

agreements. Today, we are continuing our research and development efforts in these areas with the objective to apply our MLN architecture framework to developing flexible and rapidly reconfigurable networks based on a multilayer construct.

SUMMARY

As traffic demands continue to increase, network operators will be searching for methods and techniques to more efficiently utilize their network infrastructures. Simply adding more bandwidth at a single technology layer will not likely keep up with future demands or provide a cost efficient method for network upgrades and performance improvements. The techniques described in this article are motivated by a belief that integrated multilayer control and management capabilities will be an enabling technology leading to better network resource utilization and improved user experiences. The concepts described in this article are intended to present a vision for moving forward to realize the seamless and dynamic movement of data flows, services, and virtualized network environments across all layers of network infrastructures.

ACKNOWLEDGMENTS

This research was supported in part by the U.S. DOE Office of Science under award numbers DE-FG02-06ER25741, DE-AC02-05CH11231, and DE-SC0001229. The authors are very grateful to DOE for its generous support. In addition, the authors are also very thankful to Dr. Thomas Ndousse, William Johnston, and Mary Thompson for their many insightful discussions and feedback.

REFERENCES

[1] E. Mannie, "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," RFC 3945, Oct. 2004.
[2] "Generalized MPLS Signaling — RSVP-TE Extensions," RFC 3473, Jan. 2003.
[3] K. Shimoto *et al.*, "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)," RFC 5212, July 2008.
[4] DOE Energy Sciences Net (ESNet), http://www.es.net/.
[5] Internet2 ION Network, http://www.internet2.edu/ion.
[6] USLCHNet, http://lhcnet.caltech.edu/.
[7] OSCARS: On-Demand Secure Circuits and Advance Reservation System, http://www.es.net/oscars/.
[8] C. Guok *et al.*, "A User Driven Dynamic Circuit Network Implementation," *Proc. Distrib. Autonomous Network Mgmt. Sys. Wksp.*, Nov. 2008.

BIOGRAPHIES

TOM LEHMAN [M'04] is a computer scientist in the Computer Networks Division at the University of Southern California's Information Sciences Institute (ISI). His research interests are in the areas of advanced network architectures, network control planes, end-to-end application performance, network monitoring, and network virtualization. He received his B.S. dnegree from Virginia Tech and a M.S. degree from The Johns Hopkins University in electrical engineering.

XI YANG [S'01, M'05] received a B.S. degree in communication engineering and an M.S. degree in communication and information systems from the University of Electronic Science and Technology, China, in 1997 and 2000, and a Ph.D. degree in computer science from the University of Nebraska-Lincoln in 2004. He worked with Lucent Technologies as a member of technical staff in Bells Labs, Beijing, China, in 2000. He joined the University of Southern California, Information Science Institute (USC/ISI) as a research associate in 2004. He is currently a computer scientist at USC/ISI. His research interests include advanced network architecture, next-generation Internet, optical networking, large-scale network testbed and application, network virtualization and cloud computing.

NASIR GHANI [SM] is currently an associate professor in ECE at the University of New Mexico. His research interests include cyber-infrastructure design, survivability, applications, and telehealth. He has published over 130 refereed journal and conference papers, and several book chapters, and has co-authored various standardization drafts. Overall, his research has been funded by some key U.S. government agencies (including NSF, DOE, DTRA, and NAVSEA) and some industrial sponsors. In particular, he received the NSF CAREER award in 2005 to support his work in multidomain network design. Prior to joining academia, he spent over eight years in industry and held key technical positions at several large companies (Nokia, Motorola, IBM) as well as some startups (Sorrento Networks, Array Systems Computing). He is actively involved in a range of service roles and has served as a symposium co-chair for IEEE GLOBECOM, IEEE ICC, and IEEE ICCCN. He has also chaired the IEEE ComSoc Technical Committee on High Speed Networks (TCHSN) and established the High-Speed Networking Workshop for IEEE INFOCOM. He is an Associate Editor for *IEEE Communications Letters* and has served as a guest editor for *IEEE Network*, *IEEE Communications Magazine*, and *Cluster Computing*. He received his Bachelor's degree from the University of Waterloo, his Master's degree from McMaster University, and his Ph.D. degree from the University of Waterloo, Canada.

FENG GU received his Bachelor's degree in electronics and information engineering from Huazhong University of Science and Technology, Wuhan, China, in 2008, and his M.S. degree in computer engineering from the University of New Mexico in 2010. He is currently pursuing a Ph.D. degree under the supervision of Dr. Nasir Ghani in the

Department of Electrical and Computer Engineering at the University of New Mexico. His research interests include network scheduling, topology overlay design/traffic engineering, multilayer/multidomain networking, and wireless networks.

CHIN GUOK joined ESnet in 1997 as a network engineer, focusing primarily on network statistics. He was a core engineer in the testing and production deployment of MPLS and QoS (Scavenger Service) within ESnet. He is currently the technical lead of the ESnet On-Demand Secure Circuits and Advanced Reservation System (OSCARS) project which enables end-users to provision guaranteed bandwidth virtual circuits within ESnet. He also serves as a co-chair to the Open Grid Forum On-Demand Infrastructure Service Provisioning Working Group.

INDER MONGA [M'00] (imonga@ES.NET) is developing new ways to advance networking services for collaborative and distributed science by leading research and services within ESnet. He contributes to ongoing ESnet research projects, such as Advanced Resource Computation for Hybrid Service and Topology Networks (ARCHSTONE) and ANI-Testbed as well as to the OSCARS and Advanced Network Initiative (ANI) 100G projects. He also serves as co-chair of the Network Services Interface working group in the Open Grid Forum. His research interests include network virtualization, network energy efficiency, grid/cloud computing, and sensor networking. He currently holds 10 patents, and has over 15 years of industry and research experience in telecommunications and data networking at Wellfleet Communications, Bay Networks, and Nortel. He earned his undergraduate degree in electrical/ electronics engineering from the Indian Institute of Technology, Kanpur, before coming to the United States to pursue his graduate studies at Boston University's Electrical and Electronics Communication Systems Department.

BRIAN L. TIERNEY is a staff scientist and group leader of the ESnet Advanced Network Technologies Group at Lawrence Berkeley National Laboratory (LBNL). His research interests include high-performance networking and network protocols; distributed system performance monitoring and analysis; network tuning issues; and the application of distributed computing to problems in science and engineering. He has been the PI for several DOE research projects in network and grid monitoring systems for data intensive distributed computing. He has an M.S. in computer science from San Francisco State University and a B.A. in physics from the University of Iowa. He has been at LBNL since 1990.