

Multi-Layer Networking

An Architecture Framework

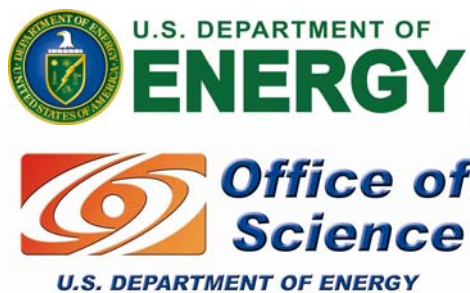


Table of Contents

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.

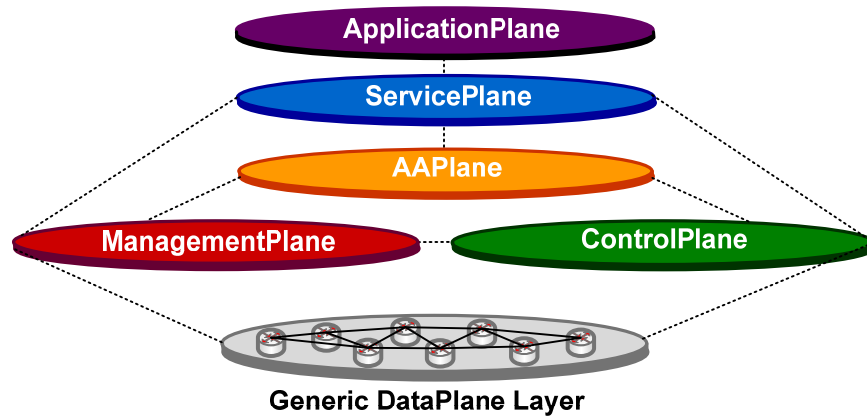


Figure 1 Capability Planes – Graphical View

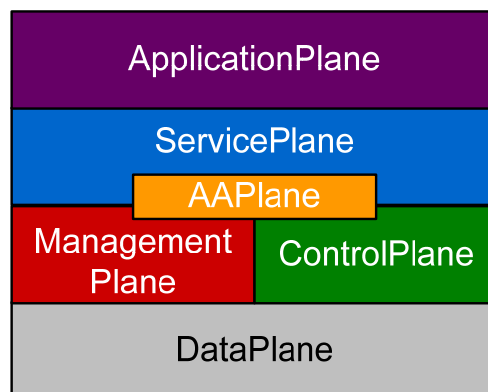


Figure 2 Capability Planes - Layered View

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.

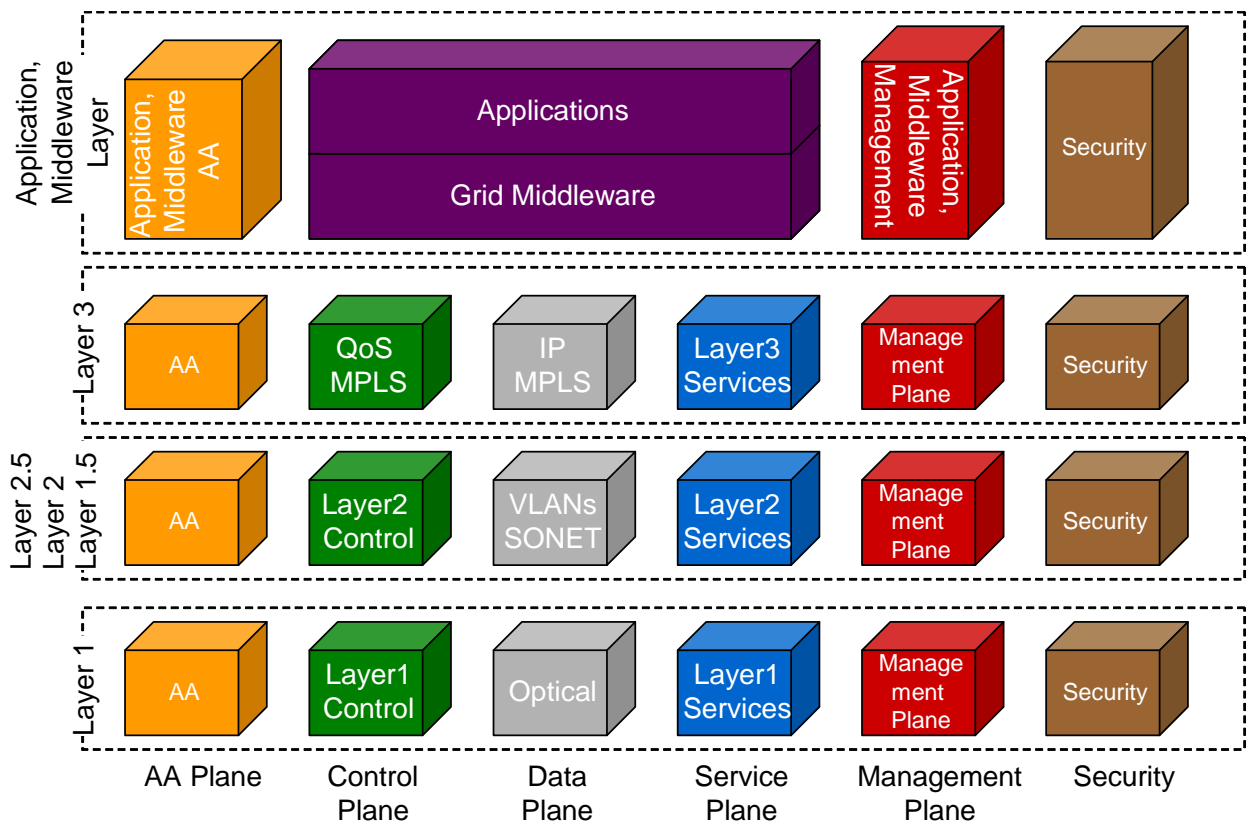


Figure 3 Layer View of Capabilities

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.

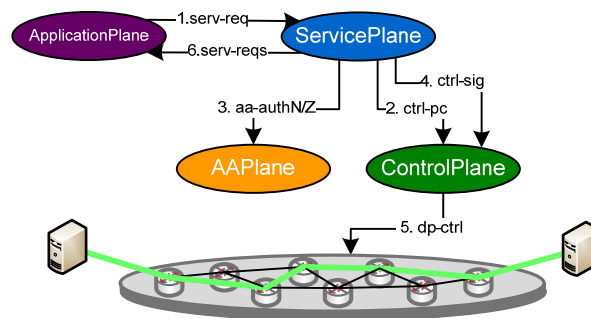


Figure 4 Single Domain Circuit Request

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.

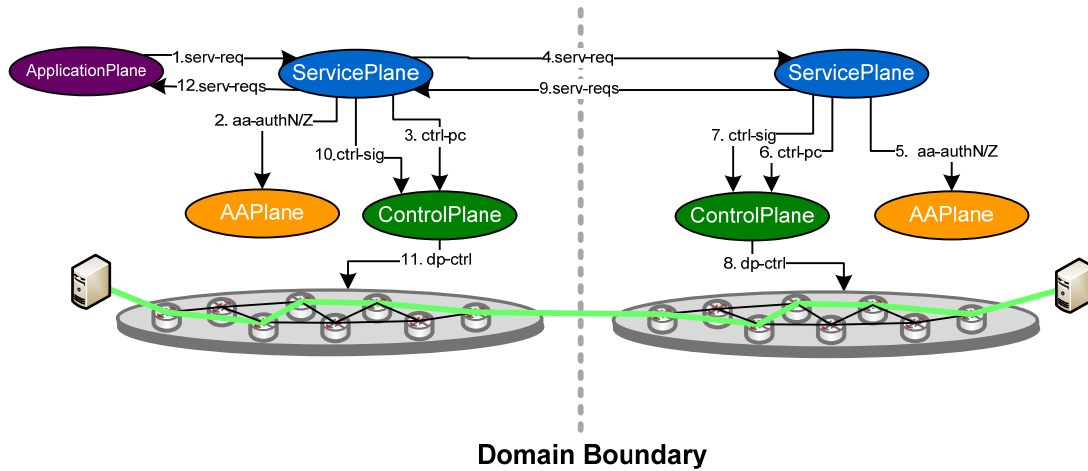


Figure 5 Multi-Domain Circuit Request - ServicePlane InterDomain

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.

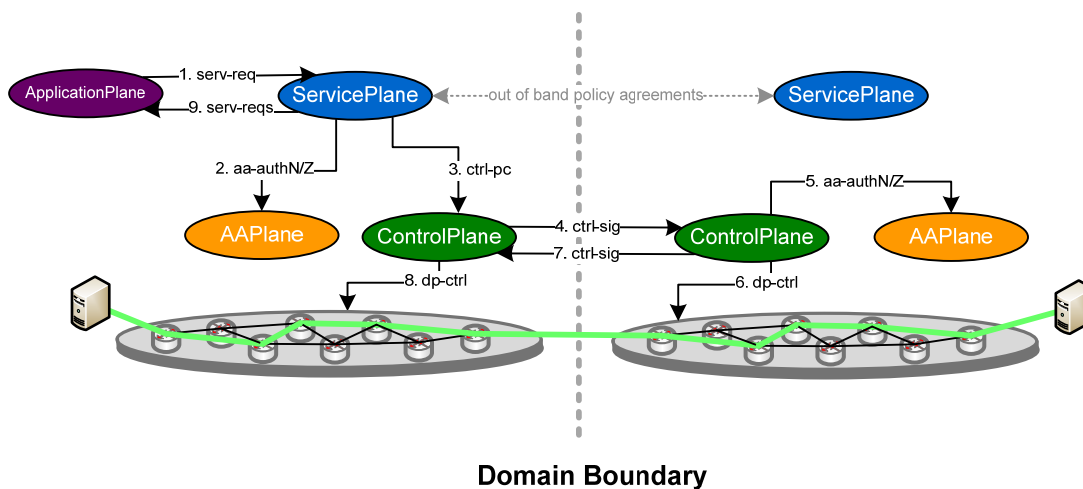


Figure 6 Multi-Domain Circuit Request - ControlPlane InterDomain

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.

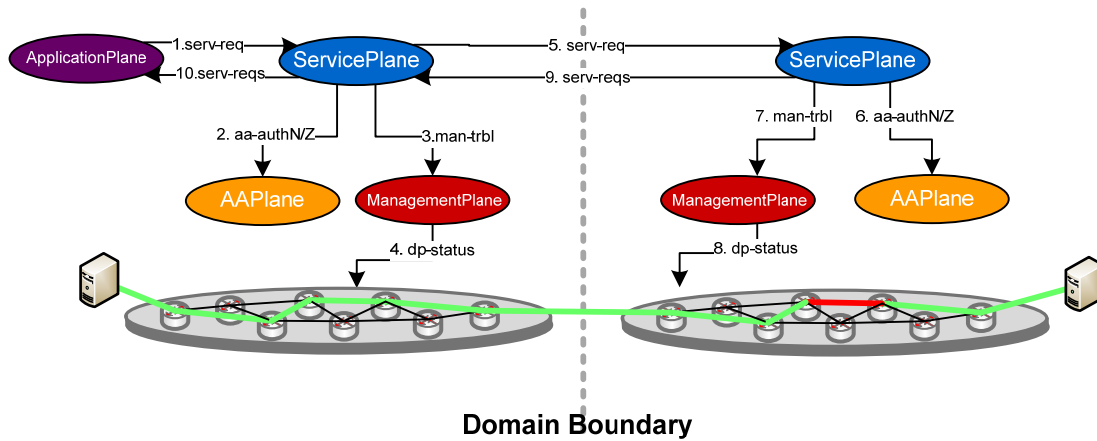


Figure 7 Multi-Domain Monitoring and Troubleshooting Procedures

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.

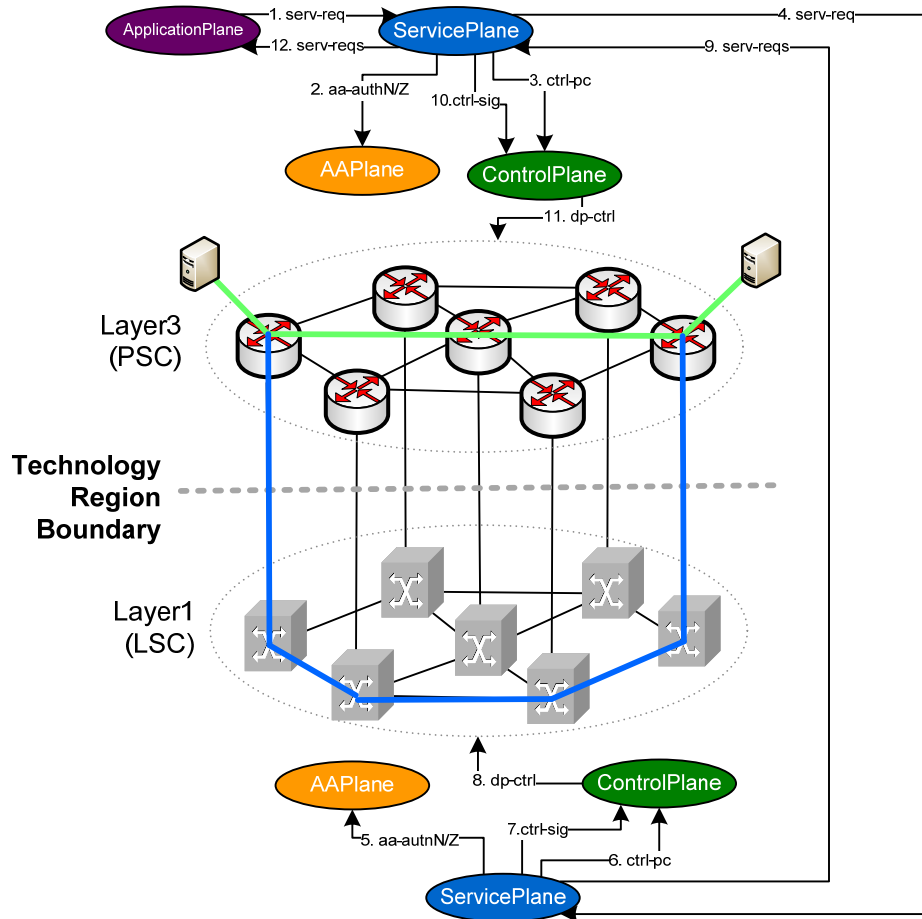


Figure 8 Multi-Layer Vertical Service Provisioning

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.

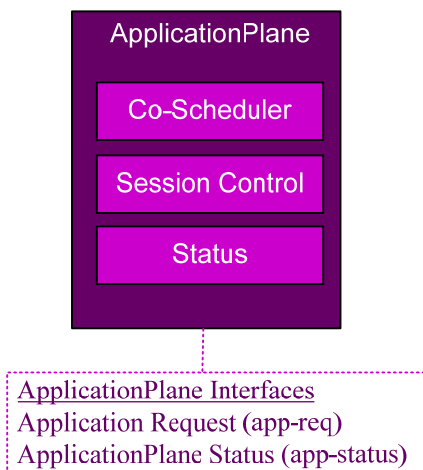


Figure 9 ApplicationPlane Functions and External Interfaces

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.

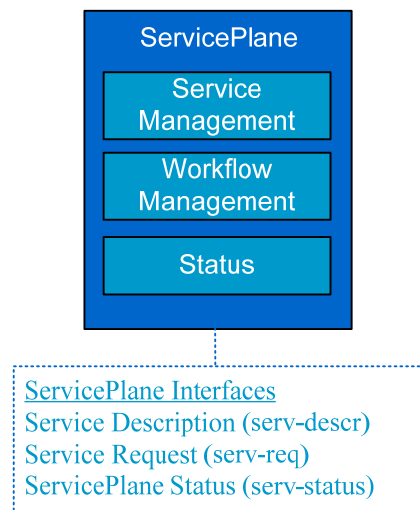


Figure 10 ServicePlane Functions and External Interfaces

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 be employed such as IPSec[2], firewalls, and access control lists.

5 AAPIane

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 AAPIane 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 latter 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.

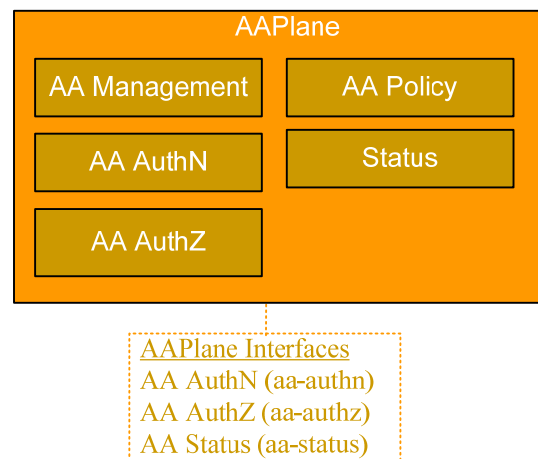


Figure 11 AAPlane Functions and External Interfaces

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.

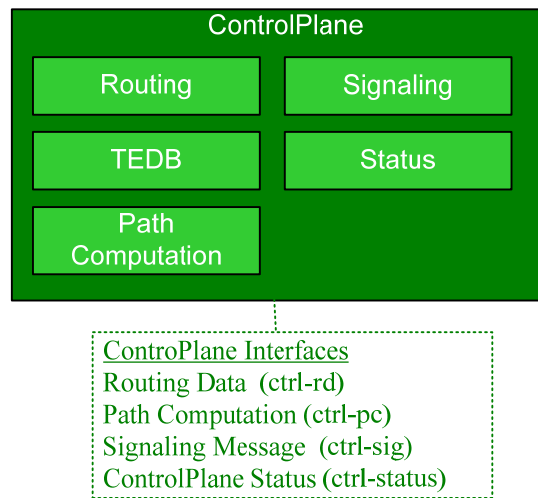


Figure 12 ControlPlane Functions and External Interfaces

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.

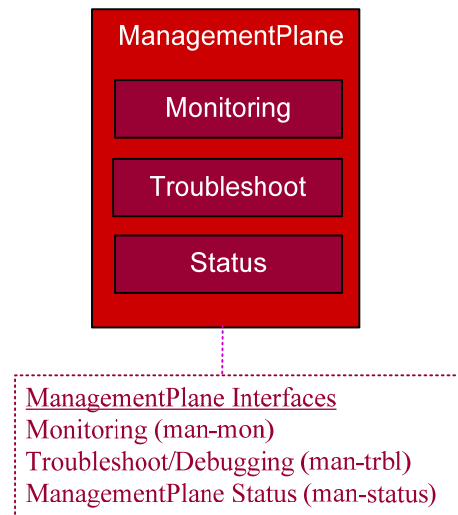


Figure 13 ManagementPlane Functions and Interfaces

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 the 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.

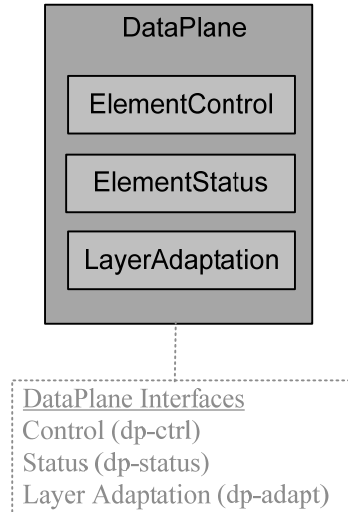


Figure 14 DataPlane Functions and External Interfaces

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.

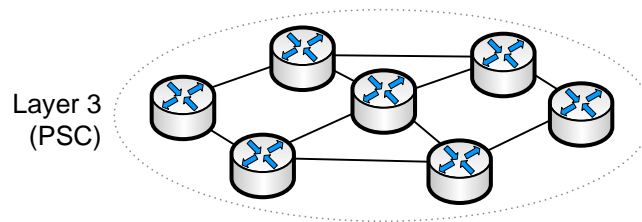


Figure 15 PSC DataPlane Type

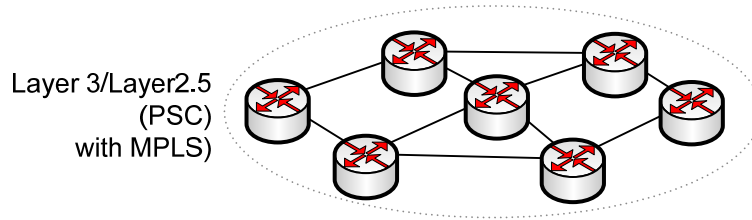


Figure 16 PSC with MPLS DataPlane Type

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.

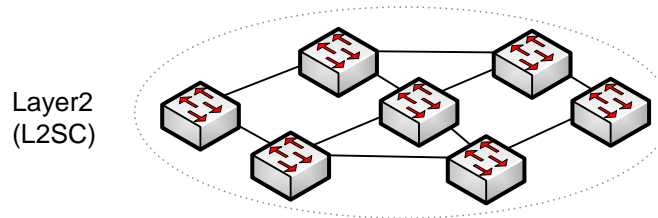


Figure 17 L2SC DataPlane Type

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.

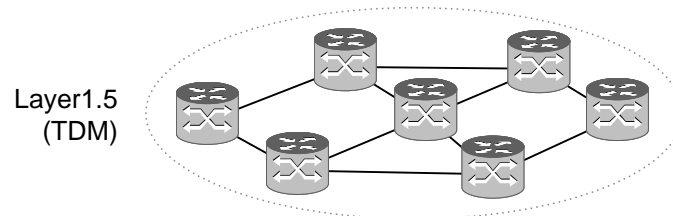


Figure 18 TDM DataPlane Type

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.

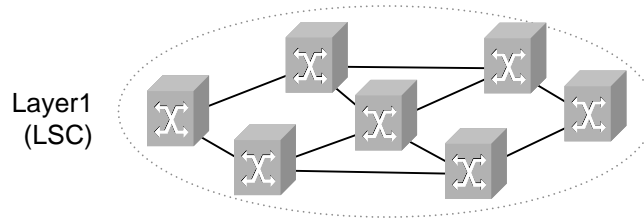


Figure 19 LSC DataPlane Type

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.

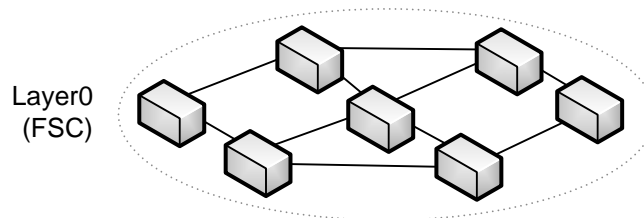


Figure 20 FSC DataPlane Type

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.

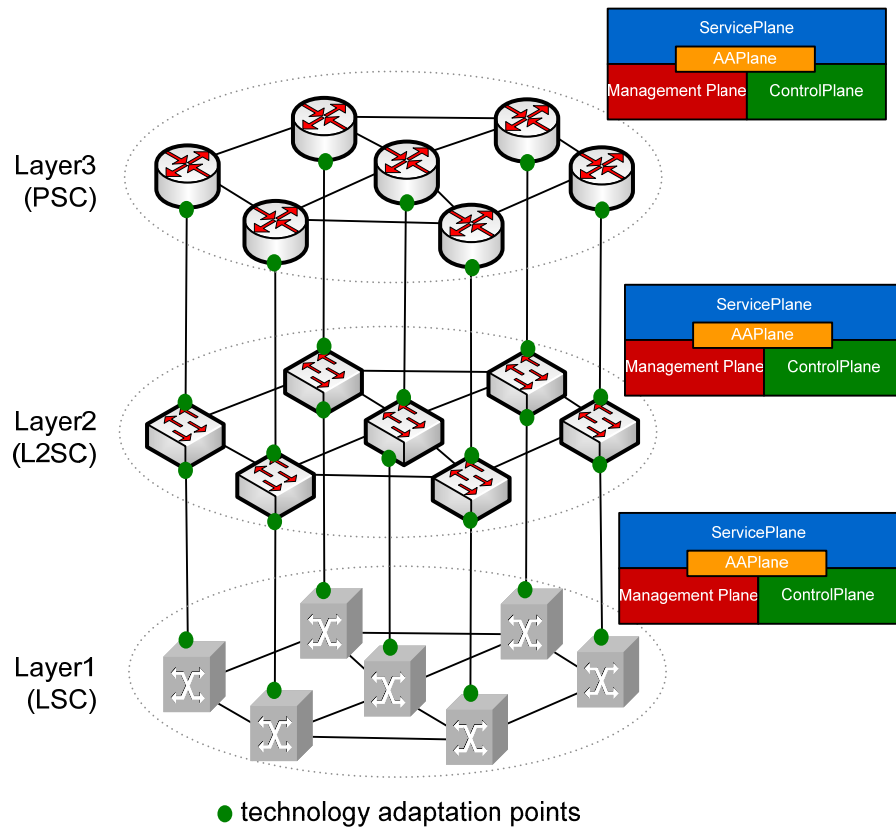


Figure 21 Multi-Layer Networking - Vertical

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.

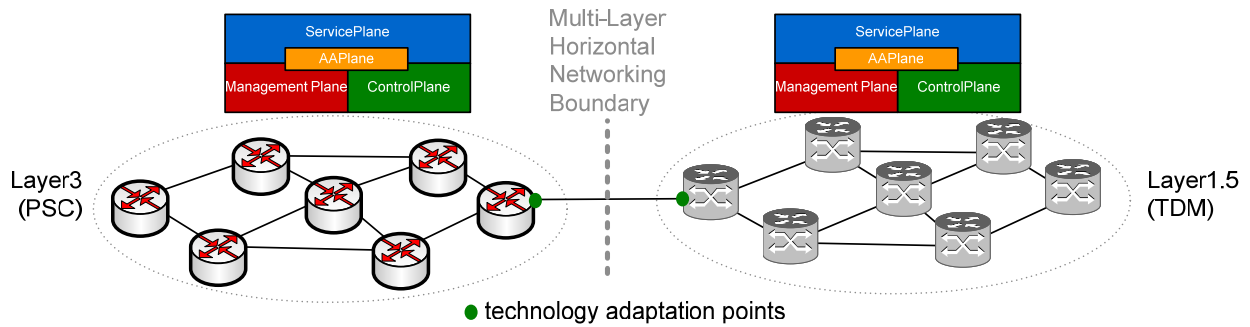


Figure 22 Multi-Layer Networking - Horizontal

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.

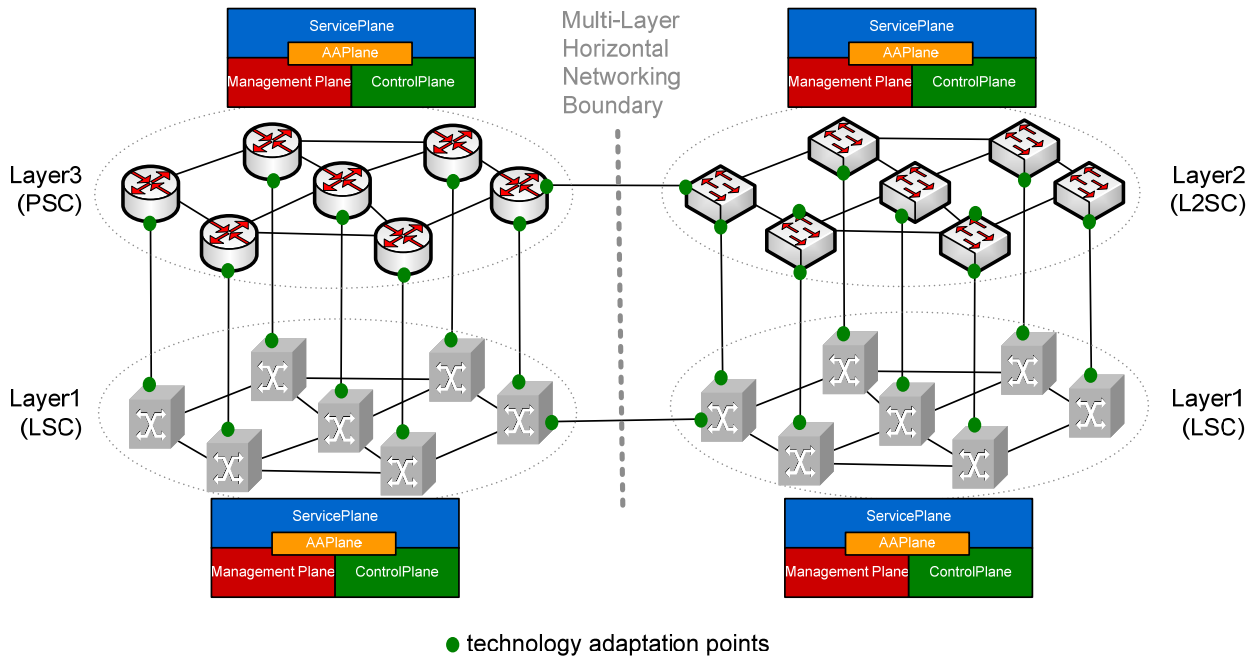


Figure 23 Multi-Layer Networking - Combined

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/>