ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, *Vol 9(10), DOI: 10.17485/ijst/2016/v9i10/84944, March 2016* ISSN (Online) : 0974-5645

A Review on Deployment Architectures of Path Computation Element using Software Defined Networking Paradigm

Sravan Yadav Eadala* **and V. Nagarajan**

Faculty of Engineering and Technology, Department of Computer Science and Engineering, SRM University, Kattankulathur – 603203, Tamil Nadu, India; sravanyadav51@gmail.com

Abstract

In traditional Generalized Multi-Protocol Label Switching (GMPLS), wide variety of functionalities like, GMPLS signaling GMPLS routing and link management increased the computational complexity of a single GMPLS network node. In concern with this, in order to provide the best paths by concerning effective utilization of network resources and better quality of transmission, a dedicated Path Computation Element (PCE) has been introduced. Software Defined Networking (SDN) is a new paradigm that decouples the network control from the data plane. With Software Defined Networking, the design, build and manage of networks can be made cost effective and dynamic manner, by transforming the traditional networks into open and programmable networks. The objective of this paper is to provide different architectural models of path computation element using Software Defined Networking paradigm. Using these deployment models, a network operator can manage and operate both the circuit switching and packet switching networks, thereby reduce capital expenses as well as operational expenses.

Keywords: Generalized Multi-protocol Protocol Label Switching (GMPLS), Optical Networks, Path Computation Element (PCE), Software Defined Networking (SDN)

1. Introduction

GMPLS architecture is generalization of MPLS architecture for circuit switching networks, which decouples the data plane, that forwards the traffic and the control plane that deals with signaling and routing functionalities. The protocols that had been designed to build GMPLS apply traffic engineering aspects to MPLS protocols. Of all, the point that makes GMPLS to differ from MPLS is, GMPLS supports multiple types of switching technologies like, packet-based switching, layer 2-based switching, Time Division Multiplexing (TDM) based switching, lambda-based switching and fiber-based switching. The demand for different services, effective use of network resources and better quality of transmission made the GMPLS control node difficult to compute the path of a given network which caused the GMPLS architecture to decouple and have a dedicated Path Computation Element (PCE).

Definite type of user traffic must be delivered with certain quality by considering different factors like, resource availability, topology information, certain link attributes and network impairments. In general, a path is defined as sequence of service provider network resources each of which are used to provide certain service. Path computation is defined as the process of finding and choosing the paths either at the time of or ahead of service provisioning[1]. The path computation element is expected to determine one or more optimal paths that have desirable probability to set up the service which is operable during the failure of certain network resources.

Software Defined Networking (SDN) is a new network paradigm that separates the network data plane that forwards the traffic using programmable switches, from the control plane, that operates the network traffic through the network controllers according to high-level policies[2].

* *Author for correspondence*

SDN is characterized by five fundamental attributes viz, plane separation, centralized control, network automation, network virtualization and openness[3]. The scope of the PCE can be extended by implementing different architectural models using SDN paradigm, which makes the network operators to directly program the PCE by employing SDN controller, thereby allowing global visibility of network topology, effective utilization of network resources and global network optimization in dynamic manner[2].

The scope of the paper is as follows: in Section 2, along with MPLS, architectural components of GMPLS and its related protocols are described. In Section 3, traditional path computation element functionalities and algorithm followed by suitable constraints that can be imposed in PCE in order to determine the optimal paths have been discussed. In Section 4, a comprehensive study on SDN and its related views are explained and followed by different architectural models of deploying SDN and PCE as well as extending GMPLS LMP functionalities as an SDN application are described, realizing the SDN paradigm for GMPLS networks.

2. Generalized Multi-Protocol Label Switching (GMPLS)

In this section, initially, MPLS technology is explained, then followed by detailed study of various building blocks and related protocol of GMPLS viz, GMPLS signaling, GMPLS routing and Link Management Protocol (LMP).

MPLS is a data forwarding technology which is being used in packet switching networks that relies on a unique identifier called label which is used by each router to find the next hop for the data packet. MPLS has its bases in IP packet switching technologies, which is the process of advancing the data packets based on tag associated with each packet. The network nodes of MPLS are called Label Switching Routers (LSR). LSR holds a look-up table, Label Forwarding Information Base (LFIB) that allows determining the next hop of the data by mapping of {incoming interface, incoming label} to {outgoing interface, outgoing label}. The packet is labeled at the source node of the LSP, which is also called ingress node, is followed by the stable mapping and arrived to the destination node, which is also called egress node. MPLS functionalities are extended to GMPLS and Figure 1 represents these building blocks and their functionalities.

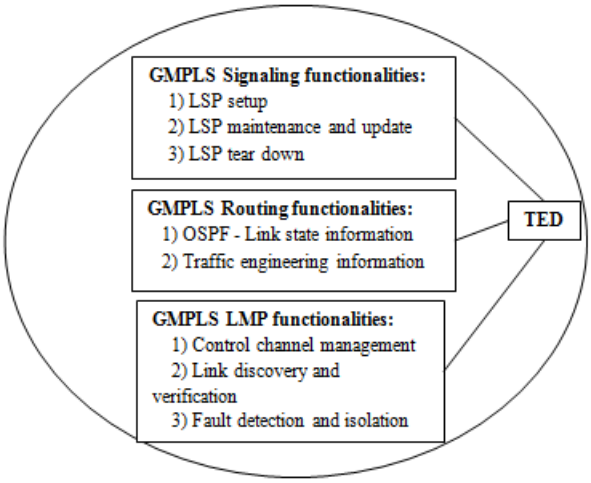

Figure 1. GMPLS functional blocks.

2.1 GMPLS Signaling

GMPLS separates its network to data and control planes. GMPLS networks use any of the two protocols for signaling, they are, RSVP-TE or Constraint-based Routing Label Distribution Protocol (CR-LDP). But, IETF discontinued extending the latter, so the GMPLS signaling protocol that is explained here is RSVP-TE[1]. The basic building blocks of GMPLS signaling are LSP establishment, LSP modification; LSP maintenance and LSP tear down[4]. In order to perform these signaling functionalities on the data plane, the control messages and processing rules are exchanged in the control plane using software components called signaling controllers. In GMPLS, each of these signaling controllers are responsible to handle the data switches, which are called Label Switching Routers (LSRs) and each signaling controller can manage its respective data switch or may manage more than one data switch. The adjacent signaling controllers are communicated in the control plane through the control channels, which are either physical or logical links between the signaling controllers. The state of all the connections of GMPLS node i.e., LSP origin, termination or traversing through a node as well as reserved resource, is maintained in the LSP database (LSPDB), which is local to itself. The GMPLS signaling messages allows each node to update its LSPDB repository.

End-to-end service in GMPLS is executed using LSP, which is used to transfer the data. Each service is accompanied by one or more LSPs. An LSP is recognized by an IP address of the sender and a 16-bit LSP ID. In order to establish the LSP in the data plane, GMPLS

signaling mechanism must identify the list of links and nodes that are to be used, which can be IP addresses. GMPLS signaling uses a signaling protocol, RSVP-TE, that is used to establish LSPs within the data plane via signaling messages in the control plane, which are carried in IP datagrams and there are eight different messages, as follows, LSP Setup, LSP Accept, LSP Confirm, LSP Upstream Error, LSP Downstream Error, LSP Downstream Release, LSP Upstream Release and LSP Notify.

Ingress LSR takes lead to establish LSP by sending an LSP Setup message to its adjacent LSR. Till then the downstream LSR accepts the request, the LSP is not established. The downstream LSR forwards the LSP Setup message to the next LSR and supplies the label that is used to reserve the resources. The LSP Setup message is forwarded to all the downstream LSR hop by hop till it arrives at the egress LSR. The egress LSR acknowledges its upstream LSR using LSP Accept, which is traversed back to all the upstream LSRs till it reaches the ingress LSR. Finally, by the time, the LSP Accept message is received by the ingress LSR, it is ready to transmit the information.

After the LSP is setup, it must be preserved till the service is no longer needed or further if any failure originates within the network, it must be rectified immediately. The ability to update an existing LSP is another feature of GMPLS signaling protocol, which may be exercised during the modification of service parameters, quality of transmission parameters, etc. Another important feature of GMPLS signaling protocol is 'make-before-break'[5], which replaces the old path with new path, thereby the old LSP is collapsed and this mechanism is useful to re-route the LSP. There are two mechanisms used to trash down the LSP, using LSP Release message that is initiated either by downstream LSR or by upstream LSR. In former case, the ingress LSR that requested the LSP sends LSP Release message. All the nodes in the data plane remove the affiliated LSP and the control plane disposes the state of the LSP, as the teardown message advances. The similar mechanism can also be initiated by the egress LSR in the reverse direction.

Most of the transport network connectivity considers bidirectional LSPs that share protection and restoration in each direction. One way to establish the bidirectional LSPs is by setting up two unidirectional paths independently in opposite directions. But this approach will have latency in establishing the LSPs, control messages overhead and resource allocation. To overcome these drawbacks, a single set of control signaling messages are used to setup the bidirectional LSPs, by allocating two different labels[4]. When a couple of bidirectional LSPs, sharing common resources and with same identifiers, moving in opposite directions, wants to establish the LSPs, contention occurs. One mechanism to resolve this contention is to give the chance to setup LSP for the higher node ID, also, issue label allocation failure message so that the latter node must try to allocate a different label and it must wait until the resources are free to use.

2.2 GMPLS Routing

With GMPLS routing, the information that is distributed across the network is gathered, which is used to direct the LSPs in the network. GMPLS routing concerns with the traffic engineering information which is used to find the optimal paths, dynamic provisioning of services and effective utilization of resources. GMPLS routing information along with traffic engineering aspects are used to determine the paths that will have the desired quality of service. Before establishing the LSP, the path along which the LSP has to be setup must be calculated, which requires the routing information that is distributed across the network, that contains the state of the link and the cost of forwarding the information through the router's interface onto the link, as well as traffic engineering information. IP routing protocols that are extended for GMPLS routing information distribution are Open Shortest Path First (OSPF) and Intermediate System to Intermediate System (IS-IS) protocols. Since each router advertises the state of all the links, these protocols are called link state routing protocols. In this paper, the former, OSPF protocol is discussed.

OSPF, developed by network working group of the IETF, is an Interior Gateway Protocol (IGP) which distributes the routing information, by flooding, between the routers belonging to a single autonomous system. Each router maintains a database called Link State Database that represents the router's current state. All the routers carry out the same algorithm and each of them builds its own tree of shortest paths with itself as root. The basic OSPF algorithm is described below[6]:

- Initially, each router sends the OSPF 'Hello' messages which are used to detect the adjacent routers.
- Then, each router advertises its link state, using Link State Advertisements (LSAs), in regular time intervals.
- LSAs are flooded across the network, which helps in synchronizing the databases of all the routers, by accumulating the LSAs and ensures that all the routers will have the same link-state database.

- Finally, each router, itself as root, calculates a shortest path tree from its database information.

The five OSPF packet types defined in RFC-2328 are, 'Hello' packet, that discovers or maintains the neighbors, 'Database description' packet, that briefs the database contents, 'Link State Request' packet, that requests the information of the neighboring database when a router needs to have up-to-date database, 'Link State Update', the local router floods the LSAs, which contain routing, metric and topology information, to its adjacent routers and 'Link State Acknowledgement', used to acknowledge each LSA that is received by the router. The GMPLS routing considers the functionality of these routing protocols along with another requirement, that is, traffic engineering.

The process of directing the traffic that furnishes the desired quality of transmission on pre-computed paths in the network is called traffic engineering. In GMPLS, the network resources that are usable for path computation of traffic engineering are formed as TE links. All the TE links and the affiliated information together form a database, which is used to compute path, called Traffic Engineering Database (TED). To setup an LSP across the GMPLS networks, along with the link state information following traffic engineering information must be distributed[1]:

- Router address, router's identity, used to send control messages to another router in the control.
- Local interface IP address, advertising router's interface address that represents the link.
- Remote interface IP address, the remote interface's address that is at the other extreme of the TE link.
- Traffic engineering metric, employed in path computation which allocates dissimilar weights to the normal and the TE traffic.
- Maximum link bandwidth, highest quantity of bandwidth that can be utilized by traffic on the link.
- Maximum reservable bandwidth, amount of bandwidth that may be reserved on the link.
- Link type, used to differentiate between point-to-point and multi-access TE links.
- Partner router, for point-to-point links, this is the router's identity that is at the other end of the TE link.
- Interface Switching Capability (ISC) descriptor, provides link switching capability for those paths of appropriate for specific service type.
- Data encoding type, describes the specific format on the links that accepts the user data.

Along with these messages, other messages, viz, unreserved bandwidth by priority, link protection type, that defends the links from acceptable susceptibility, Shared Risk Link Group (SRLG), a 32-bit unique identifier that composes set of links which share network resources whose failure may influence every link in the group and administrative group, represented by a 32-bit number that is prevented or compelled to use the links, are included in traffic engineering link information.

2.3 GMPLS Link Management Protocol (LMP)

An independent association between couples of network nodes is called a channel. These can be either data channel or control channel. The former carries only data information where as the latter carries signaling, routing and other control information as messages. In general, GMPLS network nodes are connected by many data links, each link may include many channels and each channel corresponds to certain switching capability. The increase in the number of channels and dynamic changes in the network may lead to magnificent operational cost. The Link Management Protocol (LMP) is used to maintain these overheads. Following are the functional units of LMP[7]:

- Control channel management
- Link discovery and verification
- Fault detection and isolation

LMP is carried over User Defined Protocol (UDP) using port 701 and must be assured the reliable delivery of messages. The reliability is achieved by implementing a retransmission timer at the sender. The timer is set when the sender transmits the message and if the acknowledgement is not received within that span of time, the sender will retransmit the message[1].

The LMP control channel service starts when the sender initiates a 'Config' message using the Control Channel Identifier (CCID) and bears the configuration parameters. The receiver responses with a 'ConfigAck' message, providing its own identifiers and by agreeing the parameters. If the receiver does not want to accept, it will send a 'ConfigNack' message. If the messages are transmitted both by the sender and the receiver at the same time, then as explained in the previous section, the congestion-aware mechanism is initiated and the node with the highest node identification gets the chance. The control channels regularly exchange the 'Hello' messages, to check for the active connection. If either the sender or the receiver fails to receive the 'Hello' message, within the interval of time, corresponding node will announce that the control channel is no longer alive and it prevents sending the 'Hello' messages.

Link discovery mechanism is used to determine the connectivity of the data links between the peers and link verification mechanism is used to determine the status of the links. Both these mechanisms will have similar procedures. In order to set up LSP, the status of the links must be known and LMP link verification technique determines this and provides exactly which link must bear the LSP. The link verification technique initiates process by considering the 'BeginVerify' message and the node at the other end replies with either 'BeginVerifyAck' or 'BeginVerifyNack'. Then, the sender node transmits 'Test' message and waits for the response. Then, the receiver node checks the payload and if the 'Test' message is in the payload, it will send a 'TestStatusSuccess' message, else the receiver node transports to the successive nodes and performs the same operation. If no response within the given span of time is received, then the receiver node will reply with 'TestStatusFailure', which makes the link unusable. The sender node reacts accordingly and transmits 'TestStatusAck' message, which makes the receiver node to restart the timer. Finally, the peers end the mechanism by 'EndVerify' and 'EndVerifyAck' messages.

Another significant functionality of LMP is fault detection and isolation, which is useful in transparent optical nodes that distribute the signals without analyzing. This technique is initiated by the downstream node, which sends 'ChannelStatus' message and the upstream node receiving this message determines from its adjacent upstream node whether acceptable signal is being received or not. If the upstream node finds good signal, it will return a message to its downstream node telling that the link is fine to use, else it reports the adjacent upstream as well as downstream nodes about the status of the information. Apart from these functionalities, other couple of operations, LMP includes are link capabilities exchange, that is used to inform about certain characteristics of the data links and authentication to verify peer LMP, that checks the node's identification.

3. Path Computation Element (PCE)

A path is an arrangement of vendor's network resources that can realize the particular service, which directs certain type of user traffic with better quality[1]. In legacy GMPLS networks, the path is computed by considering the traffic engineering affiliated information with certain quality of transmission constraints. In 2006, IETF has encouraged to provide a dedicated network element, Path Computation Element (PCE), which is initially implemented in MPLS networks and extended to GMPLS networks, that is committed to compute the path by formally specifying the architecture and protocol for the same. By considering such an exclusive functional component furnishes following advantages[1]:

- Network agents dynamically control and operate their networks.
- Network operators can apply their own computational algorithms and policies.
- Arbitrary customization and updating of the network is possible.
- The computational complexity overhead is reduced in a single GMPLS node.

In this section, traditional path computation in legacy GMPLS networks has been explained, followed by certain constraints that can be imposed to determine optimal paths is discussed, then architecture for dedicated path computation is explicated and the PCE protocol that is used as a communication protocol between PCE and path computation client (PCC) is specified.

Path computation is a process of calculating the optimal paths by considering the desired traffic engineering characteristics over the network resources of service providers[8]. This can be either offline or online, in the former case, the path is determined prior to service provisioning and in the latter case, the path is calculated and selected for the service at the time of provisioning. Generally, GMPLS networks consider online path computation case[1]. Also, another category of path computation process is either centralized or distributed. In case of centralized path computation, all the paths are calculated by a single node and in case of distributed path computation; there exists many collaborated computing components that issue the path. Every GMPLS network can be visualized as a Graph (G) with 'v' vertices and 'e' edges, each vertex represents a GMPLS node and each edge represented a bidirectional link with certain weights on them. Lesser the weight on the edge, optimal the path is. With these considerations, there are many computational algorithms that calculate the best path, viz., single source shortest path, all pairs shortest path, k-shortest paths, etc.

In case of GMPLS networks, the shortest path needed not be the path that is calculated just by considering the cost on the edges, rather certain additional link attributes and constraints are required to be considered and imposed to compute the optimal paths. Now, the

path computation process is represented as the constraint based path computation element that considers a vector of multiple link attributes and constraints that are regarded as a computational function. The link attributes, as a part of traffic engineering information, are already discussed in the section III.B. A path computation algorithm need not consider the entire link attributes; rather it can consider only the required properties. When the signal is transmitting from source to destination, the aggregation of physical impairments may affect the optical signal and makes it to reject at the receiver's end. So, the PCE must consider required level of quality of transmission. Besides, Optical Signal Noise Ratio (OSNR) is the major parameter that directly affect the Bit Error Rate (BER)[8]. To assure the acceptable BER, OSNR must remain at acceptable value.

The most signal quality degradation can be represented by the following equation[1]:

$D_{imp} = \Sigma f(P, B, \lambda, A_e)$

where,

P = signal attenuation; B = Bandwidth or bit rate

λ = Channel wavelength; A_e = link attributes

Following are some other physical impairment that affects the optical signal quality[9]:

- Signal attenuation: When an optical signal is transmitted through a node, there exists some power loss in the form of absorption of light. This signal attenuation is considered to be independent of wavelength. The standard SMF-28 fiber enforces about 0.25 dB/km power loss[8].
- Amplified Spontaneous Emission (ASE): One mechanism to minimize the signal attenuation is to periodically retrieve the optical signal by amplification. This process brings in added noise called ASE noise.
- Dispersion: Widening of optical signals is called dispersion and if it exceeds the threshold, neighboring bits may interfere each other. This exists in many forms and the major contributors are, Chromatic Dispersion (CD) and Polarization Mode Dispersion (PMD). CD is a process in which different wavelengths of the optical signals traveling with different velocities arrive the destination at different times. The standard SMF-28 fiber has CD value of 18 ps/(nm-km). PMD is a phenomenon in which two different polarized waveguides that are traveling at the same speed, travel at different velocities due to random spreading of optical signals. The standard SMF-28 fiber considers maximum PMD value as 0.1 ps/$\sqrt{}$km.
- Cross talk: This represents an aggregate effect of other optical components, like optical filter, wavelength multiplexers, wavelength cross-connects, etc over the path. One way to handle the cross-talk is to decrease the OSNR.

Finally, as said earlier, the Path Computation Element (PCE) must consider this signal degradation equation along with the other required constraints on the physical impairments as a vector of evaluation function and impose on the path computation algorithm that provides optimal paths with acceptable quality of transmission.

The architecture of PCE proposed by IETF is to decouple the path computation functionality from the GMPLS node and consider as a dedicated PCE node with clearly defined protocol[10]. This PCE considers two primary components, Path Computation Client (PCC), which requests the path computation to a PCE and PCE server, that considers the PCC requests, computes the path and responses the PCC, which is showed in Figure 2.

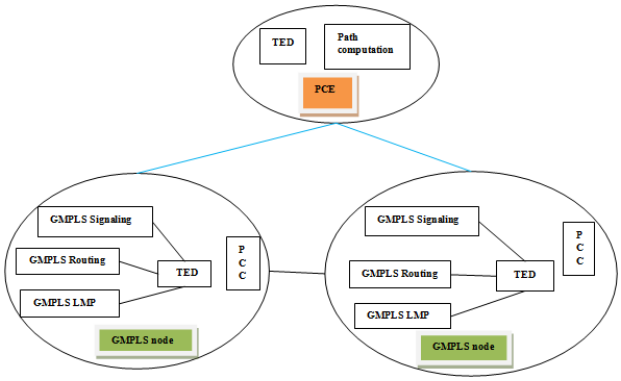

Figure 2. Path computation architecture.

As showed in the Figure 2, the PCE server contains Traffic Engineering Database (TED) that gathers the network state and TE information. One key characteristic of PCE is synchronization of TED to perform path computation. Certain designs of PCE does not store the current state of information, such a type of PCE is called Stateless PCE, which may lead to blocking of certain connections. In contrast, a Stateful PCE computes the path by considering the current state of network information from TED as well as LSPDB. A stringent synchronization of TED and LSPDB must be taken care among the GMPLS nodes and PCE.

The Stateful PCE learns the LSP state when there is an update in the network and it is called passive Stateful PCE. However, with another type of Stateful PCE, called active Stateful PCE, the path modifications and re-routing of an LSP can be performed. Besides, an active Stateful PCE that can establish or release new LSPs is called active Stateful PCE with instantiation capabilities[8]. The state of all the connections of GMPLS node i e, LSP origin, termination

or traversing through a node as well as reserved resource, is maintained in the LSP database (LSPDB), which can also be included in PCE element and the architectural diagram is as showed in Figure 3.

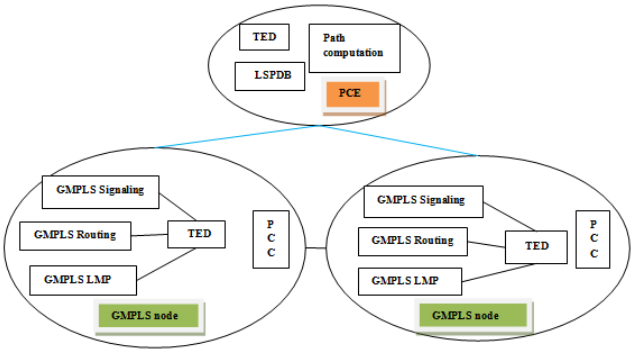

Figure 3. Active Stateful path computation architecture.

The path computation module is responsible for calculating the paths based on the path computation algorithms and constraints, also, it acts as the interface that deals with the communication protocol, PCEP, which is standard and flexible protocol that uses client and server model. Following are the essential messages used by the PCEP that are defined in RFC-5440[11].

- TCP three-way handshake messages
- Path Computation Request (PCReq) message
- Path Computation Reply (PCRep) message
- Open message
- Keep-alive message
- Path Computation Notification (PCNtf) message
- Path Computation Error (PCErr) message
- Close message

Initially, the PCEP establishes the session using TCP three-way handshake messages, 'SYN' or 'ACK' and 'Open' message is used to exchange session parameters. Then, the PCC sends a 'PCReq' message by specifying a set of link attributes and constraints in order to compute the path. After successful path computation, the PCE server replies with 'PCRep' message with combination of all the paths using an Explicit Route Object (ERO). Finally, PCEP terminates its session using 'Close' message and also TCP session.

4. Software Defined Networking (SDN)

In traditional network architecture, the network control plane and the network data plane are coupled, which makes the network rigid towards programming, no scope for global visibility of network and makes difficult to introduce new services. The new network architecture that shall overcome the above mentioned challenges and is being encouraged by certain organizations like, Open Networking Foundation (ONF), IETF, etc. is Software Defined Networking (SDN). The architecture that is proposed decouples the network data plane, that forwards the traffic using programmable switches, from the control plane, that maintains and operates the network flexibly using network controllers. The five fundamental attributes of SDN are[3]:

- Plane separation, that says about the separation of the forwarding plane and the control plane.
- Centralized controller, network control and management software.
- Network automation, the centralized SDN controller allows an open interface that automates the control of the network.
- Network virtualization, SDN provides three types of abstractions, network state, configuration and forwarding.
- Openness, important feature of SDN that considers the interface well documented and not proprietary.

SDN has wide scope in the applications like, data-centers, backbone networks, enterprise networks, Internet exchange points, etc. In this section, the functional components of SDN architecture is discussed, followed by first implementation specifications of SDN, OpenFlow is explicated, then different SDN controllers available are described and finally the integration of PCE and SDN deployment models are explained.

Principal architectural components of SDN includes SDN application, SDN controller, SDN datapath, SDN Southbound Interfaces (SDN SBI) or SDN control-data plane interfaces and SDN northbound interfaces (SDN NBI) or SDN application-control plane interfaces[8], as showed in Figure 4.

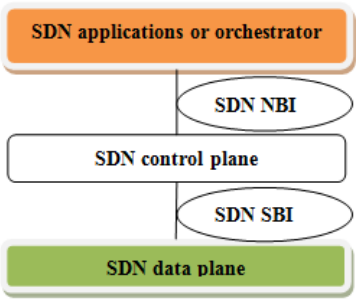

Figure 4. SDN architecture.

SDN data plane contains the network devices that have data forwarding capabilities, in regard with the

control plane instructions. SDN control plane contains centralized entity that contains the logic to control and operate the network. On the top of SDN control plane, all SDN applications are implemented which can directly programmable the communication network via NBIs. SDN SBI that is specified between SDN control and SDN data plane provides forwarding operations, statistics reporting, advertising capability and event notification. Finally, SDN NBI that is defined between SDN applications and SDN control plane provides abstractions to network view and behavior which enables the operator to manage the network through high level of policies.

There are many SDN controllers available and many considerations to look at to select a good controller. Some considerations are choice of programming languages, SBI or NBI policies, purpose and support. Following are some of them are explained.

- Network Operating System (NOX), first generation SDN controller which is open source with a couple of flavors based on the programming languages that are used to implement, either C++ or Python. Using NOX, the control programs like registering network events, topological changes, packet arrivals, etc can be written.
- Python enabled NOX is 'POX' that support OpenFlow protocol and easy to write control programs. This controller is widely used in experimentation and demonstration purposes.
- Another open source Python controller is 'Ryu' that supports various versions of OpenFlow specifications. Ryu also supports OpenStack, which is open-source cloud computing software platform. With Ryu controller, the main disadvantage is its performance.
- The open-source Java based SDN controller is 'Floodlight', which is forked from Beacon controller. It supports wide NBIs as REST API and also integration of OpenStack.
- The controller that is active in supporting of SDN and industry-wide accepted is 'OpenDaylight' that supports with network functionalities and OpenStack integration.

Apart from these SDN controllers, there exists many other open-source or commercial controllers based on their network functions, like Frenetic, FortNOX, Fresco, OSCARS, FlowScale, etc.

4.1 PCE and SDN Deployment Models

By integrating PCE and SDN controller, considering different deployment models, along with global view of network topology, global optimization of network, effective resource utilization and direct programming the network state can be achieved. Following are the various structural models that can be considered.

4.1.1 Stateless PCE and SDN Deployment Model8

In this model, the PCE does not consider state information of the network, rather it considers only topological and traffic engineering information, using TED to determine the optimal paths. Also, this model considers couple of architectures, PCE that is collocated in the SDN controller or PCE that is implemented external to the SDN controller, as showed in Figure 5 and Figure 6. In the former case, the computational complexity on SDN controller may increase. To overcome this and make the SDN controller simple, in the latter model the Stateless PCE can be considered external to the SDN controller and can be one of the applications of SDN controller.

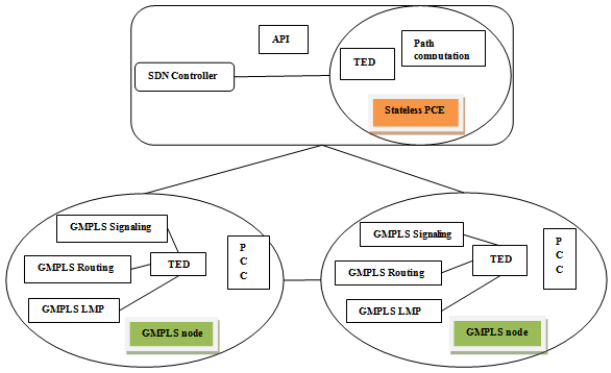

Figure 5. Stateless PCE integrated with SDN controller.

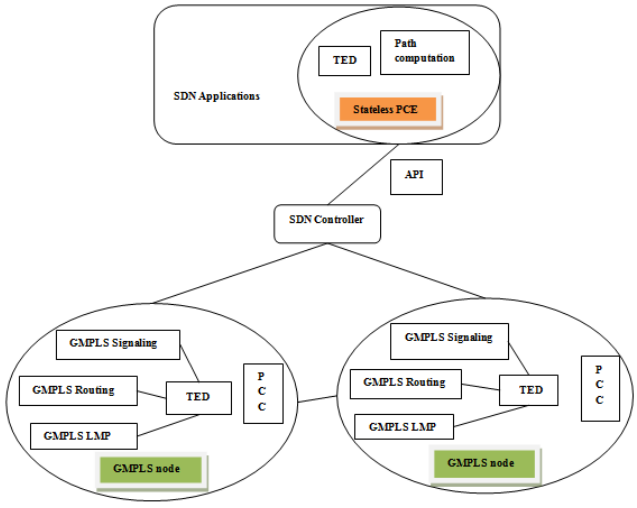

Figure 6. Stateless PCE as an SDN application external to SDN controller.

4.2.2 Passive Stateful PCE and SDN Deployment Model

Since the Stateless PCE does not consider the state information of the network, it may sometime lead to blocking of certain connections. The Stateful PCE considers state information of the network as well as the topological and traffic engineering information to find optimal paths, using LSPDB and TED, respectively. Similar to the deployment models of stateless PCE, in these models, the Stateful PCE can be either integrated in the SDN controller or PCE or PCE that is implemented outside the SDN controller, as showed in Figure 7 and Figure 8.

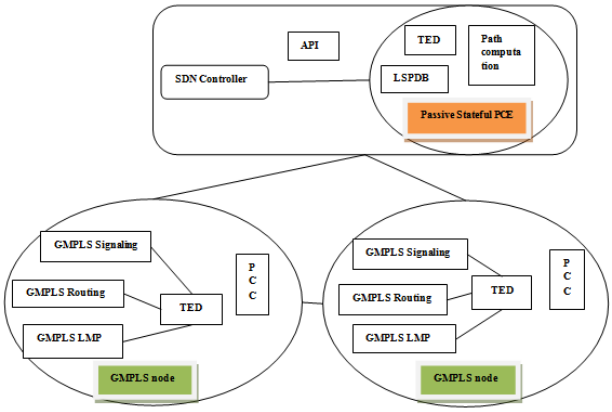

Figure 7. Passive Stateful PCE integrated with SDN controller.

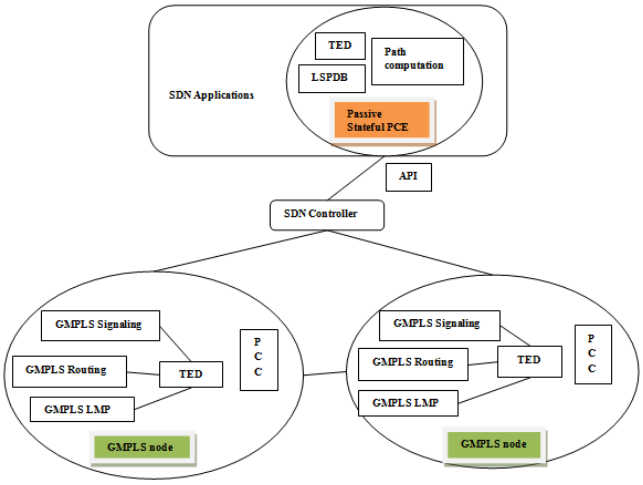

Figure 8. Passive Stateful PCE external to SDN controller.

4.1.3 Active Stateful PCE and SDN Deployment Model

If a Stateful PCE performs the signaling operations over LSP, say LSP setup, LSP modifications, LSP re-routing and LSP release, then it is called Active Stateful PCE. In this model, the Active Stateful PCE considers state information of the network, the topological and traffic engineering information as well as the LSP signaling functionalities to determine the optimal paths.

Besides, in order to point the contention that is explained in section III. A, the Active Stateful PCE must made to inform this and restore from the contention, during LSP setup phase. Again in this there are two deployment models, the Active Stateful PCE can be either integrated in the SDN controller or PCE that is implemented remote to the SDN controller, as showed in Figure 9 and Figure 10.

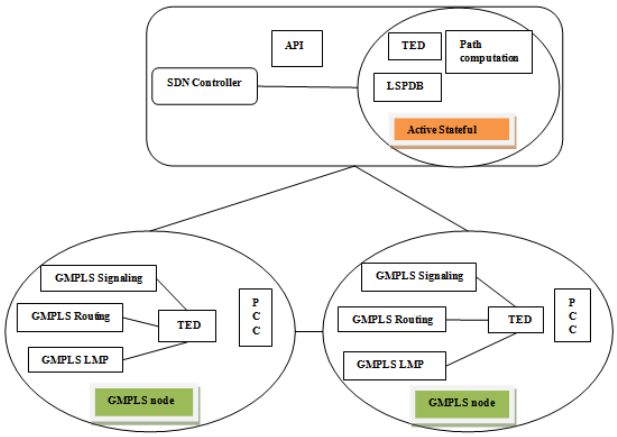

Figure 9. Active Stateful PCE integrated with SDN controller.

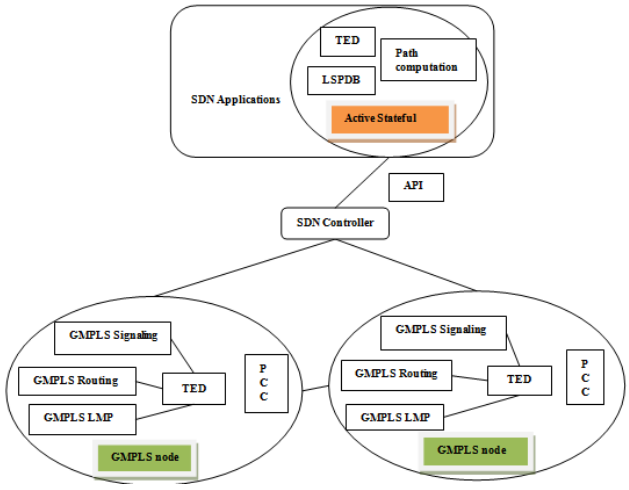

Figure 10. Active Stateful PCE external to SDN controller.

4.1.4 Extending LMP Capabilities to SDN Application

The legacy functionalities of GMPLS LMP, control channel management, Link discovery and verification and fault detection and isolation, can be extended as SDN application, as showed in Figure 11 realizing SDN paradigm for GMPLS networks.

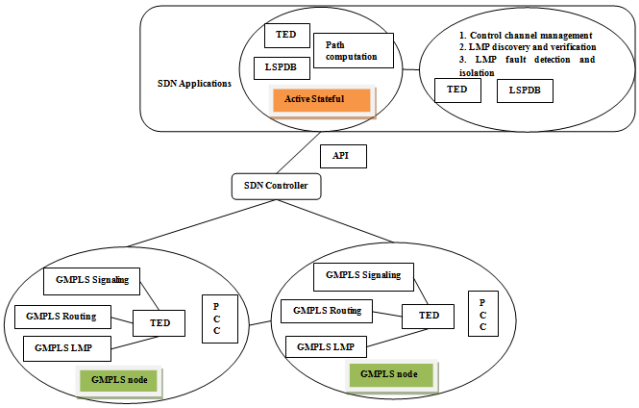

Figure 11. Active Stateful PCE and LMP functionalities as SDN applications.

5. Conclusion

This paper discusses the paradigm of SDN by considering different scenarios of PCE, Stateless PCE, Passive Stateful PCE and Active Stateful PCE. Starting with the legacy data forwarding technology, MPLS, then the need for non-packet forwarding technology, GMPLS technologies are described, here. The functional blocks of traditional GMPLS technology, GMPLS signaling, GMPLS routing and GMPLS LMP, along with their operations are explicated. After that the traditional PCE with supporting algorithms as well as the constraints that can be imposed on PCE, topological and traffic engineering, to get optimal paths are discussed. Then, the need to migrate SDN, along with SDN architecture, OpenFlow specifications and different SDN controllers are explained. Finally, the legacy architectures of PCE, Stateless PCE, Passive Stateful PCE and Active Stateful PCE, deployed on SDN paradigm, are discussed briefly. Finally, another architecture that can also be realized is extending the GMPLS LMP functionalities as an SDN application. Thus, depending on certain factors like network capacity and capability, type of network, complexity of the network, etc the required deployment model can be considered, in order to Reduce Capital Expenses (CAP-EX) and Operational Expenses (OP-EX).

6. Acknowledgement

This work has been done at Tejas-SRM SDC lab, SRM University, Kattankulathur, India and the authors want to thank the management and staff of both SRM University and Tejas Networks for their kind cooperation.

7. References

1. Farrel A, Bryskin I. GMPLS Architecture and Applications. 2nd ed. USA: The Morgan Kaufmann Publishers; 2006.
2. SDN Architecture Overview. 2013. Available from: https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/SDN-architecture-overview-1.0.pdf
3. Goransson P, Black C. Software Defined Networks - A Comprehensive approach. USA: The Morgan Kaufmann Publishers; 2014.
4. RFC 3471 - GMPLS signaling functional description. 2003. Available from: https://tools.ietf.org/html/rfc3471
5. RFC 3945 - Generalized Multi-Protocol Label Switching (GMPLS). 2004. Available from: https://tools.ietf.org/html/rfc3945
6. RFC 2328 - OSPF version II. 1998. Available from: https://www.ietf.org/rfc/rfc2328.txt. 01/04/
7. RFC 4204 - Link Management Protocol. 2005. Available from: https://tools.ietf.org/html/rfc4204
8. Munoz R, Casellas R, Martinez R, Vilalta R. PCE: What is it, How does it work and What are its limitations? Journal of Lightwave Technology. 2014 Feb; 32(4):528-43.
9. Paolucci F, Cugini F, Giorgetti A, Sambo S, Castoldi P. A survey on the Path Computation Element (PCE) architecture. IEEE Communications Surveys and Tutorials. 2013 Nov; 15(4):1819-41.
10. RFC 4655 - A Path Computation Element (PCE) - Based Architecture. 2006. Available from: https://tools.ietf.org/html/rfc4655
11. RFC 5440 - Path Computation Element (PCE) Communication Protocol (PCEP). 2009. Available from: https://tools.ietf.org/html/rfc5440