EMULATION TESTBED FOR DYNAMICALLY PROVISIONED OPTICAL NETWORKS

Aleksandar Kolarov[1],
Michael Rauch[1],
Brian Wilson[1]

[1]Vencore Labs,
United States od America

Abstract:
This paper presents the emulation testbed for a highly dynamic network environment requiring fast provisioning and restoration for a wide variety of bandwidth-on-demand services in IP over- optical networks. This paper builds on previously reported work on the development of the fast provisioning 3-Way HandShake (3WHS) and ROLEX restoration protocols to meet core optical network requirements.

The emulation testbed is a software implementation of a 100-node, global scale network control plane designed to investigate and validate the performance of provisioning and restoration protocols for the transport layer in core optical networks.

Keywords:
dynamic networks, emulation testbed, network restoration, optical networks.

1. INTRODUCTION

Recently, there has been major impetus for Bandwidth- on-Demand (BoD) in carrier networks, driven by the burgeoning demand for cloud computing services and a desire to reduce costs through bandwidth sharing. Applications such as file transfer, scheduled and unscheduled backup, virtual machine (VM) transfer, data fusion, etc. have spawned a large variety of potential requirements for bandwidth, latency, connection setup times, Quality of Service (QoS), etc. Broad spectrum of requirements for a highly dynamic IP services in optical networking environment can be found in [1]. Bandwidth requirements for core optical networks span several orders of magnitude, from ~ 100 Mbps to nearly a Tbps per service request, across multiple layers in the network. Service setup time requirements range from very fast subsecond setup times to scheduled services, with acceptable blocking constrained to be below 10-3. Service requirements also include stringent QoS metrics for packet loss (10-5) and latency (50-500 ms), as well as requirements for resilience against up to three failures for select services. Traffic level objectives are 20-100 Tbps network-wide, representing a carrier network of the future.
Traffic is nominally 75% IP and 25% transport services, comprised of a mix of best-effort IP and private line traffic along with highly variable BoD traffic. For the latter, scheduled services account for a total of about

Correspondence:
Aleksandar Kolarov

e-mail:
akolarov@vencorelabs.com

DOI: 10.15308/Sinteza-2017-186-193

20 % of the overall traffic, and true on-demand services account for about 30%. Thus, the environment is one where the familiar best effort IP and private line services dominate network traffic, and where on-demand service traffic is a relatively modest component, but highly heterogeneous in its characteristics. The "call arrival" rate is roughly 2 per second, with holding times as short as 1s-1minute for some services, thus resulting in a highly dynamic environment compared to that found in today's networks. Finally, and importantly, service requirements in [1] need to be met within an efficient network design.

Finding a solution to this complex set of IP and circuit service provisioning and restoration requirements and network capacity constraints is extremely challenging. A description of the foundational protocols, algorithms and architectures developed to address the core optical networking problem space can be found in [2], but for convenience certain relevant aspects are summarized in the sections below.

This paper focuses primarily on a demonstration of provisioning and restoration in an Emulation Testbed running real time distributed protocols on 100 servers each representing a node/city in the core optical network [2] with the control plane traffic generated by multi-Tbps BoD services. This testbed was used for protocol performance validation at large scale. The remainder of this paper is organized as follows: Section II summarizes core optical network rapid provisioning and restoration protocols. Section III describes the Emulation Testbed architecture and experimental results. In Section IV we summarize the paper.

2. CORE OPTICAL NETWORK PROTOCOLS

In [2] we reported earlier on the development of a 3-Way HandShake (3WHS) protocol which, through simulation, was demonstrated to meet the aggressive setup time in [1] and blocking metrics for single and multi-wavelength services. This paper summarizes the protocol. Section III describes validation of its performance in a 100-node Emulation Testbed.

A. Three Way Handshake (3WHS)

The 3WHS is designed to meet the optical connection creation objectives specified in [1]. The two main DARPA program requirements the 3WHS must meet are: (i) a connection setup blocking probability requirement of less than 10^{-3}, and (ii) a connection setup time of less than 50 ms + round-trip fiber transmission delay.

As defined in [3] and [4], many carriers are deploying, or are considering deploying, GMPLS with RSVP-TE to configure their optical networks. The 3WHS can be viewed as an extension of the RSVP signaling for GMPLS with the following features designed address core optical network requirements in [1]:

- "One-shot" connection setup using simultaneous cross- connections across Network Elements (NEs) to meet setup time requirements;
- Use of near real-time state data to minimize blocking;
- Evaluation of multiple paths to balance load across NEs and accommodate network failures;
- Efficient use of wavelengths and transponders to reduce network costs;
- Reducing cost of state distribution caused by short inter- arrival times.

In [5] we show that blocking under a GMPLS RSVP signaling approach is 2 to 4 times greater than when using 3WHS. These results apply when using RSVP extensions including Suggested Label and Label Set. The gap increases as network load increases.

For a given source/destination A/Z node pair, the 3WHS probes multiple fiber paths that are chosen based on multiple criteria including ability to meet setup time objectives and diversity. In particular, the longest path probed must allow a connection to be established that meets the setup time objectives. The candidate paths may be updated periodically. The 3WHS involves three signaling passes. The first signaling pass (Node A → Node Z) collects state data from each NE along the fiber path for each candidate path. No resources are reserved in Pass 1. When all the Pass 1 signaling messages arrive at Node Z, it runs an optimization algorithm to determine where wavelength conversion should be done, which wavelengths to use, etc., on each probed path. It selects which fiber path to use and what wavelengths to reserve on each fiber and at which nodes transponders are required for regeneration or wavelength conversion.

The Pass 2 signaling message from Node Z to Node A goes along the selected fiber path, and establishes cross-connections using the wavelengths and transponders selected by the Z node. These cross-connections are established simultaneously in all NEs. To reduce "backward blocking" (i.e., resources that were supposed to be reserved by Pass 2 became unavailable) extra resources can be selected by Node Z for Pass 2 to reserve. Simulations have shown that reserving one or two extra wavelengths greatly reduces backward blocking. When Node A

receives the Pass 2 message, it knows which wavelength connections have been successful, and makes the final decision on the resources to be used. Node A then initiates its cross-connects to the client ports, sends a notification to the client, and sends a Pass 3 message to Node Z. This message is used to release the extra reserved resources along the selected path and to inform the Z node of the specific connections used. More detail regarding the 3WHS protocol is in [arch paper] and [5].

We have adapted the 3WHS protocol to support OTN SWL traffic and for Multi-Carrier domain use. In the former case, the 3WHS collects ODU0 state information, the Z node uses slightly different optimization criteria and ODU0 channel selection is deferred until Pass 2 since there are no variable transponder costs to minimize (see [5] for more details). In the latter case, the 3WHS is structured in two levels: inter- domain probing and intra-domain probing. The A-node initiates Pass 1 signaling along multiple inter-domain paths where each carrier domain probes multiple intra-domain paths for each potential inter-domain path (see [5] for more details).

B. Restoration Signaling (ROLEX)

The core optical networks also require that connections are resilient to network failure. All connections should be restorable from any single network failure while some connections should be restorable from any 2 or 3 network failures as well (at most 1 node failure). We use a shared mesh restoration strategy coupled with a restoration signaling protocol based on AT&Ts Robust Optical Layer End-to-End X-connection (ROLEX) protocol [6]. Just after the working connection is established, our PCE computes 1, 2 or 3 diverse restoration paths for the connection (depending on the connection's resiliency requirement) and determines how many wavelengths are needed for restoration on each link of those paths. Signaling is used so that each node is aware of exactly which wavelengths are reserved for restoration on each link.

As shown in Figure 1, when a failure occurs, an Alarm (either Loss of Signal or Alarm Indication Signal) propagates from the failure site to the connection A and Z nodes. When the alarm reaches the A and Z node, the alarm is "aged" for 10 ms to model typical carrier practice of aging alarms to insure that the condition is not transient. Connection A and Z nodes both then begin ROLEX signaling on one of the connection's restoration paths. ROLEX signaling proceeds from both end nodes for each connection affected by the failure; from A towards Z and from Z towards A.

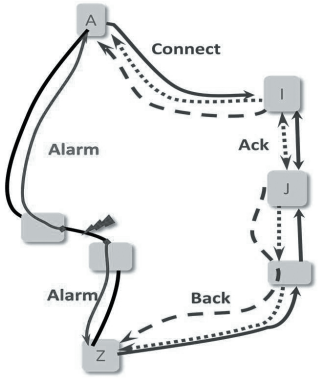

Fig. 1. Failure Model and ROLEX Signaling.

The ROLEX signaling protocol consists of Connect (orange arrows), Ack (blue dotted arrows) and Back (purple dashed arrows) messages. The Connect and Ack messages are used to establish cross-connects for connections being restored by communicating the demand for and supply of wavelengths as each connection being restored. At some point (nodes labeled I and J), the Connect messages from A and Z nodes "cross" each other which allows nodes to determine that ROLEX signaling is complete. The Back message is used to signal all that way back to the connection A/Z node that restoration has failed and that any resources used by, and connections created for, this connection should be released.

ROLEX signaling could use any available wavelength for restoration but our implementation used bandwidth that has been allocated for restoration. Reserving specific wavelengths for restoration increased the probability that ROLEX would be able avoid the use of transponders on restoration connections. Two methods are used to allocate bandwidth for restoration. First, at call setup time the 3WHS first establishes the working connection and then the PCE computes the restoration path(s) and determines how many wavelengths need to be reserved for restoration on each link of the restoration path(s). A signaling message is used to set aside specific wavelengths for restoration based on this data. Second, the PCEs periodically compute the number of wavelengths needed for restoration on each link based on the active connections in the network and signals the nodes at each end of the link to set aside specific wavelengths. The latter method is needed to release wavelengths that are no longer needed for restoration. Note that in the shared mesh restoration strategy used in our approach the same wavelength may be used to restore multiple connections with different end nodes.

3. EMULATION TESTBED

We have previously evaluated the performance of the 3WHS and ROLEX signaling protocols using Discrete Event Simulation (DES). In the work reported here, we developed a laboratory facility to verify the real-time performance of the 3WHS and ROLEX signaling protocols at scale. The testbed does not include a data plane so our experiments are designed to verify the performance of the control plan. We sought to verify that the 3WHS was able to meet setup time and blocking objectives specified in [1]. We also sought to verify that the shared mesh strategy was able to reserve sufficient bandwidth for restoration and that the ROLEX protocol was able to meet the restoration objectives.

As a side effect of the software implementation, we were able to demonstrate the simultaneous operation of provisioning and restoration. That is, we tested the performance of the 3WHS under network failures and interactions between the 3WHS and ROLEX signaling.

A. Overview

The Vencore Labs Optical Networking Emulation Laboratory consists of 110 Dell Xeon servers running CENTOS 6.2. One hundred of the servers emulate optical NEs and run software implementing the 3WHS and ROLEX protocols. Six of the servers run software implementing the Path Computation Element (PCE). An additional server runs a "Testbed Controller" that configures the emulated network topology and supplies the optical demand requests. The emulated nodes are implemented in C++ using a custom software framework designed to support the high speed processing requirements of the 3WHS and ROLEX protocols and allowed us to look at performance data at a more detailed level than would normally be available in a commercial control plane implementation.

The core optical global network consists of 100 nodes (75 CONUS, 25 Non-CONUS). The network includes 136 inter- nodal optical links, of which 13 have two fiber pairs, and the rest one. Every fiber pair supports up to 100 optical channels. The additional fiber pairs were allowed under the requirements in [1] and were used so the most demand services consistently met blocking objectives. The testbed supports 6 Path Computation Elements (PCEs), of which 4 are CONUS and 2 are non-CONUS Figure 2 shows a map of the CONUS topology of the core optical network.

Fig. 2. CONUS Network.

Figure 3 is an architectural diagram of the core optical network emulation software. Each optical node runs a copy of the Border Controller (BC) and Optical Control (OC) software which implements the 3WHS and ROLEX protocols. The Demand Generator (DG) sends messages to the BC to establish and teardown connections. Each of the 6 PCE nodes provides restoration paths for services, as well as periodically updating the candidate routes at each node used by the 3WHS for probing. PCEs also send messages to every other PCE to keep their databases in loose synchronization. The Testbed Controller (TBC) configures the testbed, including the network topology (with emulated link delays), and starts and stops the DG at the beginning and end of an experiment.

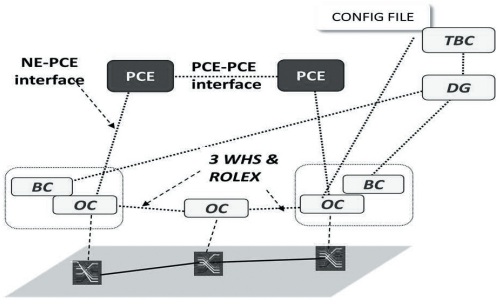

Fig. 3. Software Architecture.

B. 3WHS Experiments

An emulation experiment consists of running a demand stream representing 24 hours of traffic against the optical network. The optical network load consists of an IP network designed to support 15Tbps of IP services, and 5Tbps of optical wavelength service requests. The IP network is designed to support the offered load and all possible failure scenarios. Thus, the links supporting the IP network appear as static optical demands that

do not need to be restored following a failure. The IP network is supported by 1880 optical connections spread over 209 different A/Z node pairs. Following loading the IP network, the utilization of optical links ranged from 2 to 67 wavelengths.

The optical demand makes up 25% of the offered load and consists of requests for 1, 2, 4 and 8 wavelength connections. The optical demand is divided roughly 40% for Very Fast setup, short-holding time demands while the remaining bandwidth is for more traditional optical services allowing longer setup times and longer (almost static) holding times. This traffic mix means that more than 99.9% of optical service requests are for the Very Fast setup short holding time optical demands.

For an emulation experiment, optical demand streams are constructed by simulating several months of demand activity to obtain a random traffic state and taking a snapshot of the active demands forming a set of "embedded" demands. We then continue simulating demand stream for the next 24 hours of traffic. The demand active at the instant of the snapshot are the embedded demand and are loaded by the emulation when it starts and the remaining demands are emulated as the emulation unfolds. The arrival rate for dynamic optical demands is approximately 2.5 requests per second (about 170,000 demands per day).

A total of 51 experiments were performed resulting in more than 8.5 million service requests. The results of these experiments demonstrated that the emulated 3WHS, run against the CONUS topology and traffic matrix, resulted in blocking ranging from $4E^{-5}$ (single wavelength) to $2E^{-4}$ (8 wavelength) meeting program objectives.

Setup time objective was that it had to be either less than an absolute setup time (CONUS < 100 ms, non-CONUS < 250 ms); or a differential setup time (no more than 50 ms longer than the round-trip propagation time on the shorter of the diverse pair of paths of shortest total length between the source and sink). Figure 4 and Figure 5 show the measured absolute and differential setup times for CONUS and non- CONUS demands. In Figure 4, the red dashed line shows the CONUS objective while the blue dashed line shows the non- CONUS objective. The "step" in the non-CONUS results is due to the mix of demand pairs that have very different setup time properties in the non-CONUS set (e.g., Amsterdam to London and Denver to Delhi are both "non-CONUS" demands). More than 99% of CONUS and non-CONUS demands meet the absolute setup time objective. In Figure 5, the 50 ms objective is shown by the green dashed line. The 50 ms objective was met for approximately ~99% of service requests.

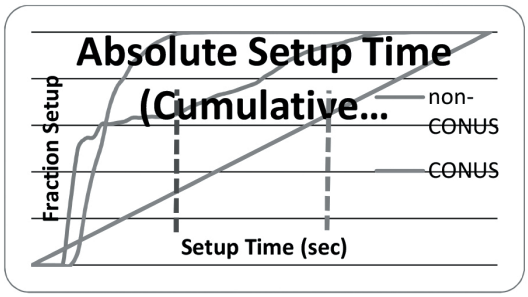

Fig. 4. Measured Absolute Setup Time.

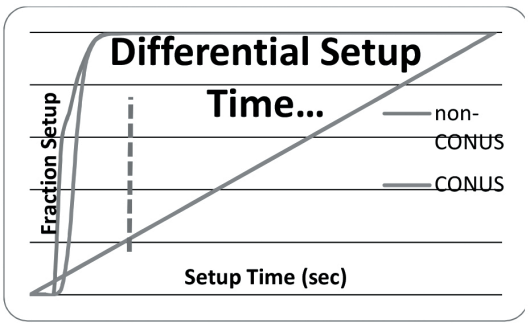

Fig. 5. Measured Differential Setup Time.

C. ROLEX Experiments

We used the emulation testbed to validate restoration in the core optical network using ROLEX. Specifically, we verified that the PCEs reserved sufficient bandwidth; that ROLEX was able to use that bandwidth to restore connections affected by failures and that the time to restore connections met program objectives. We assumed that there were an unlimited number of transponders at each node.

In an implementation note, we were faced with the problem of determining exactly when service was restored in the emulation testbed. Normally, the restoration of a connection is detected at the connection A/Z node by the presence of the optical signal from the far end. This is not feasible in the emulation since we do not have a data plane. We modeled the restoration of the client signal using a Rest message. The Rest message proceeds from the A/Z node to the Z/A node (waiting at nodes until the cross-connections are complete). The connection is restored when the Rest message arrives at the connection A and Z node.

Single Failure

The emulation software was written in such a way that every single network failure could be evaluated in a single emulation run. In this case, each demand stream is

emulated for 4 hours and then the emulation steps through all 236 node/link failures and executes ROLEX signaling for each failed connection. The software measures restoration success/failure and times. These were repeated 58 times resulting in nearly 14,000 failure events and ROLEX signaling for approximately 70,000 failed connections.

Note that at the time of a failure, the only nodes that know about the failure are those adjacent to the failure (e.g., nodes that terminate a failed link). In particular, the A/Z nodes of connections affected by the failure don't know what failed only that the failed resource was supporting the working connection. The most extreme case is where the connection A/Z node fails. In this case, the other end will attempt to restore the connection only to have the attempt fail at the penultimate node of the restoration path.

There were a small number (< 0.5% of failed connections) where ROLEX ended abnormally. These abnormalities were due to calls that failed while being setup or taken down or failed before restoration capacity was fully reserved (in which case the call was not really protected at the time of the failure). There were even instances where the connection was restored before the alarm arrived at the end point.

Like connection setup time, restoration time objectives were specified using absolute and differential time objectives. Figure 6 shows the measured absolute restoration time. More than 99% of CONUS and 94% of non-CONUS demands meet the restoration time objective. Figure 7 shows the measured differential restoration times for all CONUS and non- CONUS demands. More than 99% of demands met the differential restoration time objective. Notice that for many demands the differential restoration time is less than 0. This is due to ROLEX signaling proceeding from both A and Z nodes so the restoration time is sometime close to the *one way* propagation time between the nodes while the differential restoration time subtracts the *round-trip* propagation time.

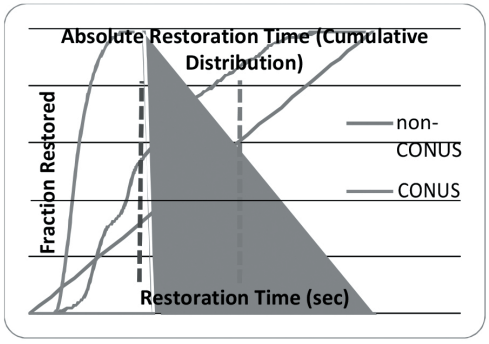

Fig. 6. Single Failure Absolute Restoration Time.

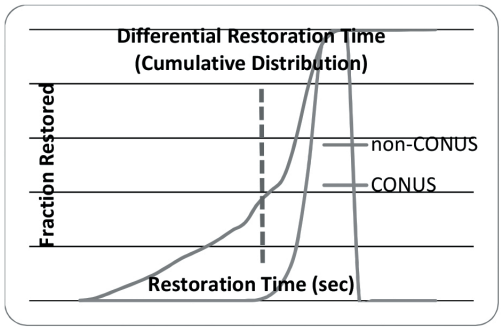

Fig. 7. Single Failure Differential Restoration Time.

As seen in Figure 6, the results for non-CONUS absolute restoration time were not as impressive, where nearly 6% of non-CONUS demands did not meet the absolute restoration time objective. This is due to ROLEX using a controlling node for each link. The controlling node selects the wavelength to use to restore a demand on a link. If the ROLEX connect message arrives first at the non-controlling node, then it may take a round trip on the link before a connection established at the non-controlling node. In this case, the restored signal may take experience a delay of up to 3 times the propagation time for a link. Within CONUS, this is not a significant issue but with long (e.g., trans-oceanic links) the delay can be significant. ROLEX could be modified to avoid this delay but it may result in a bi-directional connection being restored on two different wavelengths in the same link. This outcome may not be acceptable to some carriers.

Multiple Failures

We also used the emulation testbed to verify the performance of ROLEX under multiple failures. Unlike the single failure case, the emulation could only model one multiple failure scenario in each experiment. Coupled with the large number of multiple failure scenarios, this limitation drastically limited our ability to evaluate ROLEX for multiple failures. Given these constraints, we focused on high-stress scenarios (scenarios that nearly partition the network). Figure 8 illustrates a typical multiple failure scenario emulated using the testbed.

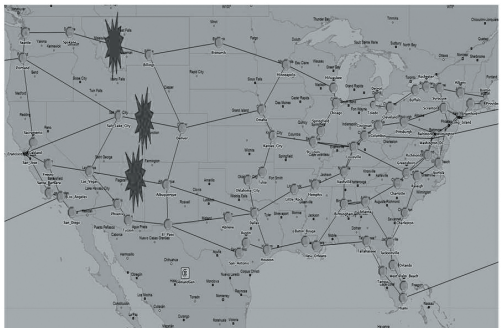

Fig. 8. Sample Multiple Failure Scenario.

Each experiment consisted of emulating demand for one hour and then failing 3 resources in succession. Enough time elapsed between failures for ROLEX signaling to complete and then all nodes were notified of previous failures. This allows ROLEX to evaluate the status of candidate restoration paths for demands resilient to 2 or 3 failures when selecting which of the restoration paths to signal along. This is a less restrictive assumption than specified in [1] which would allow us to recalculate restoration paths between failures.

Figure 9 shows the measured differential restoration times following the first (f1), second (f2) and third (f3) failures. More than 99% of demands affected by the first failure, 98% of demands affected by the second failure and all demands affected by the third failure met the differential restoration time objective. This chart is based on relatively few events (e.g., only 24 demands were restored following the third failure).

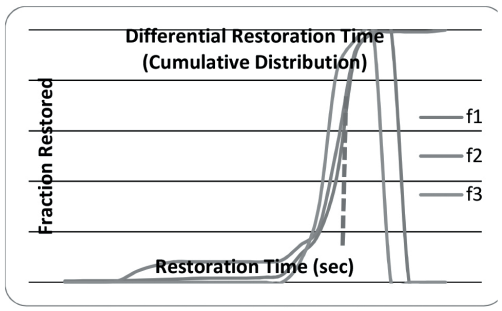

Fig. 9. Multiple Failure Restoration Time.

As with the single failure case, more than 99% of CONUS demands met the absolute restoration time while 90% of non- CONUS demands met the absolute restoration time objective.

Debugging the emulation did uncover an issue with multiple. In particular, the PCE calculation of restoration capacity assumes that following the second or third failures, only demands with those high resiliency requirements would be restored. In effect, the PCE calculations assumed that previously restored low resiliency demands could be preempted to restore a high resiliency demand. In the emulation, demands with lower resiliency requirements were not preempted and occupied some of the restoration bandwidth causing some high resiliency demands to fail in ROLEX. This was not observed in our experiments but was observed in test cases we designed with limited demand streams in limited networks.

4. SUMMARY

Dynamic optical networking, particularly in support of BoD, has been a fertile research area for many years. In recent years, there is a renowned interest in a solution for supporting BoD in a highly heterogeneous service environment with demanding provisioning and restoration targets. In this paper we presented the emulation testbed consisting a software implementation of a 100-node, global scale control plane network designed to investigate and validate the performance of provisioning and restoration protocols for the transport layer in core optical networks. We also demonstrated the behavior of the 3WHS under network failures and interactions between the 3WHS and ROLEX signaling immediately network failures. Emulation results presented in this paper confirm that the 3WHS protocol meets the aggressive setup time in [1].

REFERENCES

[1] A. A. M Saleh, "Dynamic Multi-Terabit Core Optical Networks: Architecture, Protocols, Control and Management," DARPA BAA 06-29, Proposer Information Pamphlet.

[2] Chiu, A., G. Choudhury, G. Clapp, R. Doverspike, M. Feuer, J. Jackel, J. Klincewicz, G. Li, P. Magill, J. Simmons, R. Skoog, J. Strand, A. Von Lehmen, S. Woodward, and D. Xu, "Architectures and Protocols for Capacity-Efficient, Highly-Dynamic and Highly-Resilient Core Networks," Journal of Optical Communications and Networking (JOCN), Vol. 4 (1), Jan, 2012, pp. 1-14.

[3] E. Mannie, Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture," RFC 3945, October 2004.

[4] IETF RFC 3473, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions," Jan 2003.

[5] Skoog, R., J. Gannett, K. Kim, H. Kobrinski, M. Rauch, A. Von Lehmen, B. Wilson, "Analysis and Implementation of a 3-Way Handshake Signaling Protocol for Highly Dynamic Transport Networks.", OFC 2014, San Francisco, CA, March 2014.

[6] Chiu, A., R. Doverspike, G. Li, J. Strand, "Restoration Signaling Protocol Design for Next-Generation Optical Network," OFC/NFOEC 2009, San Diego, CA, Mar. 22-26, 2009, Paper NTuC2.