Progress Report

DOE Award number: DE-SC0004909
Name of recipient: University of Massachusetts, Dartmouth
Project Title: Coordinated Multi-layer Multi-domain Optical Network (COMMON) for Large-Scale Science Applications
Principal investigator: Vinod Vokkarane
Date of Report: February 29th, 2012.
Period covered by the report: December 1st, 2011 - February 29th, 2012.

Introduction

We intend to implement a Coordinated Multi-layer Multi-domain Optical Network (COMMON) Framework for Large-scale Science Applications. In the COMMON project, specific problems to be addressed include 1) anycast/multicast/manycast request provisioning , 2) multi-layer multi-domain quality of service (QoS), and 3) multi-layer multi-domain path survivability. In what follows, we outline the progress in this quarter for the above categories.

Activities

In this quarter, our research team at University of Massachusetts, Dartmouth held several conference calls with the researchers/software developers of the ESnet team, regarding the anycast design and development issues in OSCARS. Specifically, our team proposed different schemes for the deployment of the anycast communication paradigm and consulted with the software developers at LBNL, as to which would be the more scalable and efficient design to implement, keeping in mind the further services intended to be provided.

Progress/Accomplishments

In this section we describe the progress and accomplishments in each of the tasks (labeled T1, T2) as outlined in the project proposal:

- T1: Anycast/Multicast/Manycast Request Provisioning

  With the advent of bandwidth intensive applications, the demand for multicasting/manycast networking capabilities has become an essential component of wavelength division multiplexed (WDM) optical networks. To support these functionalities in an optical network that is Multicast Incapable (MI), i.e., the optical cross connects are incapable of switching an incoming optical signal to more than one output interface, one must implement a logical overlay to the underlying optical layer. Two traffic models are usually considered for wavelength routed networks: static and dynamic. A static traffic model gives all the traffic demands between source and destinations ahead of time. Dynamic traffic requests arrive one-by-one according to some stochastic process and they are also released after some finite amount of time. We can further classify the above traffic models as immediate reservation (IR) or advance reservation (AR) requests. The data transmission of an IR demand starts immediately upon arrival of the request and the holding time is typically unknown for dynamic traffic or assumed to be infinite for static traffic. AR demands, in contrast, typically specify a data transmission start time that is sometime in the future and also specify a finite holding time. Advance reservation is also referred to as scheduled demands, especially when considering static traffic.

  Work Performed & Findings:

  In this quarter, we extended our work from the previous quarter and developed two lower bounds on the minimum number of wavelengths required to provision IR multicast request set. Note that the lower bound is not the *actual* minimum number of wavelengths required, but just a *theoretical* bound. On comparing the lower bounds with the ILPs it was observed that the ILP was within (7-10%) of this bound. This work along with the work presented in the earlier quarter was accepted for publication in a journal [1]. We have proposed extensions of the DAAN and DAMN (explained in the previous quarterly report) models to incorporate AR requests provisioning. In [2], we addressed the problem of provisioning a static set of multicast AR requests in a MI network (such as the ESnet). Our results have indicated that the DAMN

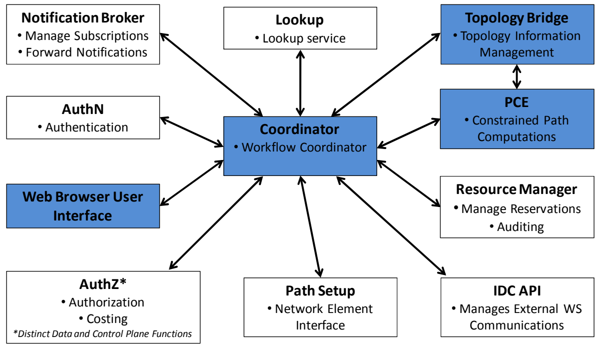

Figure 1: OSCARS modular framework.

and DAAN approaches result in a significant reduction in the number of wavelengths required to provision a static request (25-30% reduction) as compared to MVWU.

- T2: Survivability Development in OSCARS

  We have developed a new single-domain path computation element (PCE) implementation for the OSCARS 0.6 framework that protects against single-link failures by provisioning survivable requests. The new survivable PCE modules will give researchers access to reliable connections for critical data transfers.

  Design

  The PCE is responsible for computing a single path given the existing network topology, and a connection request. In OSCARS 0.6, this service is provided as a framework which allows third-party PCE implementations to be developed and deployed alongside the rest of OSCARS' modules. The four main modules involved in the path computation request flow are the user interface (or IDC API), coordinator, topology bridge, and PCE modules (shaded modules in Fig. 1).

  Like the rest of the OSCARS framework, PCEs are modules and each one is represented within the OSCARS Coordinator by a PCE Proxy that handles the communication between the Coordinator and the PCE. Requests to PCEs are assumed to be asynchronous. The PCE framework provides:

  – **Modularity:** each PCE is executed as an independent process.
  – **Distribution:** PCEs can be deployed on different (virtual or physical) hosts other than the OSCARS IDC host.
  – **Security:** PCEs follow the OSCARS 0.6 security model in regard to authentication, authorization, and accounting.
  – **Language neutrality:** while the default binding is JAVA, the APIs are based on web-services, thus allowing for independent developers to use any language as long as they comply with the API specification.

  OSCARS 0.6 allows several PCEs to be deployed, each one of them responsible for computing a specific subset of local paths in a given domain. The execution process is defined as a flexible PCE workflow module, whereby purpose-specific component PCEs are connected in a workflow graph to incrementally prune network resources that do not meet the constraints of the user or network operator. As such, the output from one module can then be fed as input to the next. Specifically, our proposed anycast PCE processes a network topology (domains + nodes + ports + links) as input and outputs a single path from the source to a selected destination.

Implementation Our proposed survivable PCE design is composed of four core modules which compute the primary and backup paths in two passes. On the first pass, the request and a network topology are taken as input, and the primary path is returned. On the second pass, the request, the primary path, and a network topology are input, and the link-disjoint backup path is returned. The two passes through the PCE stack are controlled by a new API method (createSurvivableRes) which first sends a synchronous call to the createRes API method to set up the primary path. Once information about the primary path is received, it is encoded into the request's optionalConstraints and createRes is then called a second time. As a best effort service, if a primary path is found but no link-disjoint backup path exists, then the request will be provisioned as a non-survivable request. Also, in the case where the source or destination node has only one out-going link, then the backup path is allowed to share this link with the primary.

Following the unicast model, our proposed Path Computation Element (PCE) is composed of four core modules:

- **ConnectivityPCE**: This PCE module is responsible for computing the network topology corresponding to the network connectivity graph between the source node and the destination node. The output of this module is an updated topology with node-pairs not physically connected by a physical fiber pruned out. This module is responsible for dynamically interpreting the network domain so that all other PCEs do not improperly assume additional connectivity.
- **BandwidthPCE**: This PCE removes the links, ports, and nodes that do not guarantee the bandwidth capacity of the user's request. Fibers which are oversubscribed at the starting time of the request will be pruned from the topology. The behavior of this PCE is largely responsible for the existence of resource-driven connection blocking.
- **VlanPCE**: Each port on a node has a designated number of VLAN tags which represents the maximum number of virtual circuits which may be accommodated at that node. The VlanPCE module prunes out the links, ports and nodes that do not have enough VLAN tags to support the virtual circuit., thereby guaranteeing secure connection establishment for all successfully provisioned requests.
- **DisjointDijkstraPCE**: This PCE module computes the end-to-end path from the source to the destination. When computing the path, this PCE first checks the contents of the request's optionalConstraints. If this is empty (pass 1), then the primary path is computed as normal. If the optionalConstaints contain an encoded path from the source to the destination (pass 2), then the links used on this path are pruned from the current network before the backup path is calculated. This guarantees that Dijkstra's algorithm will either generate a path which is link-disjoint from the primary, or fail to find any path.

• **T2.1: What-If OSCARS for Service and Resource Discovery**

What-If is a multi-domain offline reservation protocol and service implementation for generating ranked viable reservation solutions according to the QoS requirements of the user that are SLA abiding. What-if provides effective solution for request blocking in OSCARS by providing alternative reservation solutions to the users and facilitates network reservation planning as well as What-If Analysis on the network, for privileged users. The impacts of the What-If architecture can be outlined as follows:

[1] Significantly reduces processing overheads in the control plane by eliminating reservation re-attempts and blind probing of the network by the user.

[2] Parallel generation of QoS based, ranked viable reservation solutions, provides faster and closer reservation matches to user requirements thereby reducing user effort and time.

[3] Service allows users to query for the best set of candidate reservation solutions, available to them at a future time without actually committing to reserve (offline).

[4] What-if service adds intelligence to a network scheduling software, to help provision, network resources for multi-constrained user requests across multiple layers and domains, more efficiently.

The proposed solution eliminates the processing overheads in the control plane caused by re-attempts at reservation due to blocked requests, prevents blind probing of the network, and reduces user effort by fetching relevant QoS based alternative/candidate reservation solutions. This work has been submitted to a conference and is under peer-review [3].

NOTE: All the above work which has resulted in conference proceedings have acknowledged the DOE-COMMON project.

NEXT QUARTER DELIVERABLES

- Extend the work in T1 to account manycasting communication paradigm for AR requests.
- Incorporate Traffic Grooming in tasks T1.
- Extend the current OSCARS framework to provision point to multipoint (manycast/multicast) connection establishments.
- Investigate issues on multi-domain QoS Provisioning.

References

[1] A. Gadkar, J. Plante, and V. M. Vokkarane, “Multicast overlay for high-bandwidth applications over optical WDM networks,” in *J. of Optical Commun. and Networking*, 2012.

[2] T. Entel, A. Gadkar, and V. M. Vokkarane, “Scheduled multicast overlay for bandwidth-intensive applications,” in *Conference on Optical Network Design and Modeling*, Essex, UK, April 2012.

[3] B. Ramaprasad and V. M. Vokkarane, “What-If Driven Multi-Constrained OSCARS Framework,” in *Proc. of IEEE GLOBECOM 2012*, 2012, under review.