DOE COMMON
2011-12 Annual Project 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: Dr. Vinod M. Vokkarane
Date of Report: May 10, 2012
Period covered by the report: June 1, 2011 – April 30, 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 the above categories (Year 2 deliverables).

The work conducted in this report was performed at the University of Massachusetts, Dartmouth in the Computer and Information Science department. Dr. Vinod Vokkarane (principal investigator) is the director of the Advanced Computer Networks Lab (ACNL), which comprises of the following research team:

Post-Doctoral Research Associate:
Dr. Arush Gadkar

Graduate Students:
Jeremy Plante (Ph.D.)
Thilo Schndienst (Ph.D.)
Bharath Ramaprasad (MS)
Mark Boddie (BS/MS)
Tim Entel (BS/MS)

Progress/Accomplishments

In this section, we describe the progress and accomplishments in each of the tasks (labeled T1-T3) as outlined in the COMMON 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. A traffic matrix is given and the goal is typically to find a routing and wavelength assignment that can meet all the demands and minimize overall cost (e.g. using the least number of transmitters/receivers). Dynamic traffic requests arrive one-by-one according to some stochastic process and they are also released after some finite amount of time. When dynamic traffic is considered, the number of transmitters and receivers is fixed and the goal is to minimize request blocking. A request is said to be blocked if there are not enough resources available to route it. 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:

As stated in the previous annual report, we addressed the problem of provisioning static and dynamic IR (multicast and manycast) requests respectively [1], [2], [3], [4]. We presented two solutions: *drop at member node* (DAMN) where we restrict the termination of a lightpath only to members of the multicast session, and *drop at any node* (DAAN) model, wherein we allow a lightpath to be terminated at any node and compared the performances of these approaches to a naive approach of establishing end-to-end lightpaths from the source node of a session to each destination member of the multicast session. We referred to this approach as *multicast via WDM unicast* (MVWU). For the static case of provisioning the multicast/manycast requests, we formulated integer linear programs (ILPs) to solve the problems with a goal of minimizing the total number of wavelengths required to service the request set. For the dynamic case, we presented heuristics to minimize the blocking probability of a request (considering that each link in the network is equipped with the same number of wavelengths).

This year, we developed two lower bounds on the minimum number of wavelengths required to provision the 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. The bounds together with the proposed models and heuristics was recently accepted for publication in the journal of optical communications and networking [5]. The corresponding work related to the manycast communication paradigm is currently in progress for submission to an IEEE journal.

We also investigated the multicast advance reservation problem in MI networks. Under the advance reservation traffic model (also know as scheduled traffic), connection requests specify their start time to be some time in the future and also specify their holding times. We have proposed extensions of the DAAN and DAMN models to incorporate AR requests provisioning. In [6], 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 and DAAN

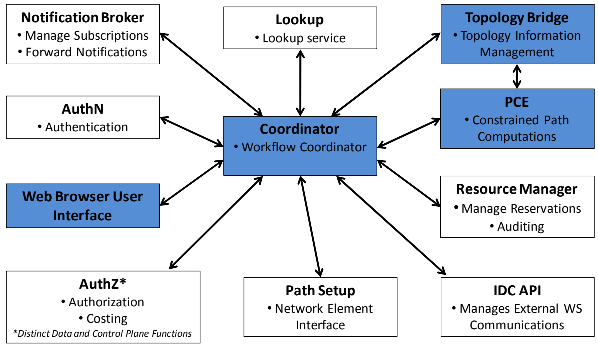

Figure 1: OSCARS modular framework.

approaches result in a significant reduction in the number of wavelengths required to provision a static request (25-30% reduction) as compared to MVWU. We have also considered a dynamic traffic scenario for the AR multicast requests and aimed at lowering the blocking probability (defined as the fraction of connections that cannot be provisioned). We have proposed efficient heuristics to solve the DAMN, DAAN, and MVWU AR models. Our proposed heuristics reduce the blocking probability (by more than two orders of magnitude) for the DAMN and DAAN models as compared to the MVWU approach. This work is currently under peer review [7]. The corresponding work for the manycast communication paradigm, i.e., provisioning manycast AR requests in a MI network is also under peer review [8]. Currently, we are in the process of developing bounds for the AR static traffic cases and also working on proving the NP completeness of these problems. This work is targeted for a potential submission to an IEEE journal.

- **T2: Anycast Development in OSCARS**

  We have developed a new multi-domain path computation element (PCE) implementation for the OSCARS 0.6 framework that takes advantage of the anycast paradigm. The new anycast PCE modules will allow researchers to execute future destination-agnostic applications over ESnet, thus broadening the number of available services, and improving the network resource utilization globally. Furthermore, we extend the OSCARS framework to not only perform intra-domain path anycast computation, but also extend such computation across different network domains managed by different instances of OSCARS for making inter-domain anycast path computation possible. Overall, our development demonstrates the feasibility of our proposed implementation and evaluates the improvement of the anycast routing over unicast.

  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

Following the unicast model, our proposed anycast PCE is composed of four core modules which take an anycast request and a network topology as inputs, and output an updated, pruned topology (refer to Fig. 2):

- **AnycastConnectivityPCE**: This PCE module is responsible for computing the network topology corresponding to the network connectivity graph between the source node and all the candidate destination nodes of the anycast group. 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.
- **AnycastBandwidthPCE**: This PCE removes the links, ports, and nodes that do not guarantee the bandwidth capacity of the user's anycast request. Fibers which are over-subscribed 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. Our testing results, show how the probability that requests will be blocked is reduced as an effect of utilizing anycast communication.
- **AnycastVlanPCE**: 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 *AnycastVlanPCE* 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.
- **AnycastDijkstraPCE**: This PCE module computes the potential end-to-end paths to each destination in the anycast set and then selects the final destination based upon

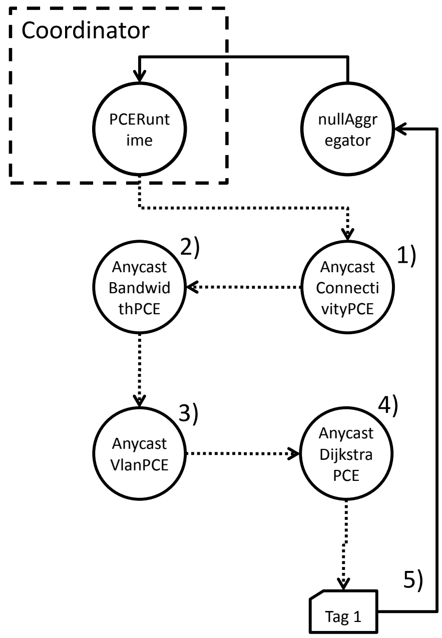

Figure 2: Anycast PCE stack flow-chart.

some criteria. In this work, we select destinations to satisfy an anycast request such that the candidate along the shortest path is preferred. Alternative metrics can easily be incorporated with our existing *AnycastDijkstraPCE* design to select destinations based on a path's available bandwidth, and/or other metrics.

The worst-case runtime complexity of our anycast PCE implementation is increased over its unicast counterpart by a factor of $|D_s|$, the number of destinations in the anycast set.

OSCARS is responsible for providing the understanding of inter-module relationships and the ordering of the module executions. A *PCERuntime* agent controls this ordering through a customizable XML configuration file that prescribes rules for arranging the PCE module executions. PCE modules need not be aware of the relative execution ordering. The *NullAggregator* module aggregates a set of paths based on the result from several PCEs. In our case, the *NullAggregator* captures the result, *Tag 1* (refer (5) in Fig. 2), from the last PCE to execute, *AnycastDijkstraPCE*. The final reply is sent back to the *PCERuntime* module, which governs the request forwarding between PCE modules. The final output from the execution of the anycast PCE workflow is a pruned topology consisting exclusively of the VC along the path from the source to the selected anycast destination.

- **T2.1: Multi Domain Anycast**

The multi-domain workflow for an anycast AR request is shown in Fig. 3. For the sake of

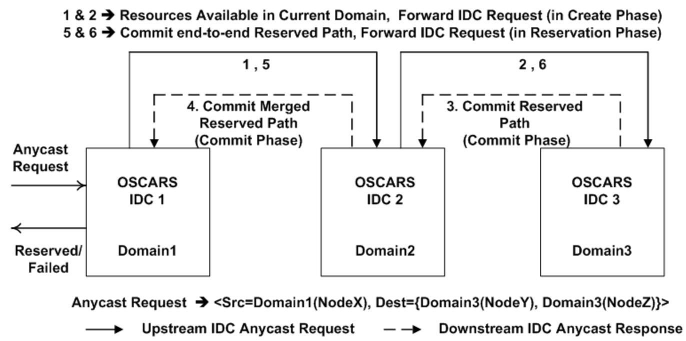

Figure 3: Multi-domain anycast in OSCARS.

simplicity, consider an IDC as a single OSCARS instance. A multi-domain anycast request is first submitted to the local IDC (source IDC is IDC 1 in this case). In this example workflow, the anycast request specifies Node X in Domain 1 which is found locally in the network managed by IDC 1. The request also specifies Node Y and Node Z as part of the anycast destination set, which is remote to IDC 1. Now the Coordinator in IDC 1 initializes the PCE workflow. The *anycastConnectivityPCE* loads all the partially (ingress and egress only) or fully visible (sister network domains share entire topology) topologies as a topology stack to reach from the source to all the anycast destination domains. In this example Domain 2 and Domain 3 are loaded. The *anycastBandwidthPCE* and *anycastVlanPCE* then prune all the local nodes, ports, links in the topology stack which do not fit the user's constraints of bandwidth and VLAN. In case of MPLS, they simply prune the ingress and the egress nodes of the local domain. This pruned topology stack is then fed into the *anycastDijkstraPCE*, which finds the best local path to the egress node for all valid anycast destinations of the local domain and returns this path to the local Coordinator. Now the local Coordinator within IDC 1 determines that the request is inter-domain, flags the anycast request to be in the CREATE phase and forwards this request by loading the profile of the next inter-domain hop in the path which helps to communicate suitably over the inter-network with the next IDC responsible for the inter-domain hop. In Fig. 3, IDC 1 forwards the inter-domain anycast request to IDC 2. Now, IDC 2 performs actions similar to IDC 1 (the OSCARS coordinator and PCE framework are highly re-entrant and efficient by switching logic based on the phase a request is in). If a local path is found feasible, IDC 2 then forwards this request to IDC 3 which manages the destination domain, Domain 3. Now, IDC 3 performs actions similar to IDC 2 in computing the best path to all of the anycast destinations and returns whichever has the shortest number of hops back to the Coordinator. Upon successful receipt of the path, the Coordinator for IDC 3 then locally saves the path in its local database as reserved and changes the anycast request phase to COMMIT and forwards the request back to the sender of the request, IDC 2. IDC 2 sees the phase of the request to be COMMIT, and so it merges the local path with the global path and saves this merged path as the reserved path in the local database. IDC 2 again forwards the updated request to the original sender, IDC 1, which after merging the local and global paths, sets the full end-to-end path in its local database. Subsequently, IDC 1 changes the request status to RESERVED to indicate the end-to-end path is stitched and forwards this end to end reserved path to IDC 2. IDC 2 now overwrites the entire end-to-end anycast path again into its local database and forwards it to

IDC 3 which performs similar action of persisting the end-to-end reserved path to the local database. IDC 3 flags the reservation as completed by setting the anycast request status to the RESERVATION-COMPLETE phase, which is then recursively transmitted back to IDC 1 (the original sender). The user is notified that the anycast AR request has been successfully reserved. If the request cannot be provisioned locally in the CREATE phase by any of the domains in the path, then the user is notified that the reservation has failed and the request is blocked.

Results

We stress tested our anycast extension to OSCARS on ESnet and GEANT networks independently for single domain performance of anycast as well as both the domains in case of testing for multi-domain. During the performance analysis, we observed that anycast performs significantly better (about 40% better) in terms of blocking on an average when compared to unicast. This performance improvement is backed with a significant reduction in average hop count (about 55% better when compared to unicast) required to provision the network demand which in turn reduces the number of lightpaths to be setup in the network significantly. In essence by extending OSCARS for anycast we were able to observe significant betterment in performance. This work encompassing the design, development and performance for both single and multi-domain was presented in [9].

- **T2.2: Multicast/Manycast Overlay in OSCARS**

  We have developed a service extension to the existing OSCARS framework which allows point-to-multipoint communication paradigms for supporting specialized requests. Applications taking advantage of this new service might incorporate data replication, backup storage, using the network as a data cache, etc. Traditional multicasting/manycasting involves the delivery of a request from a single source site/server/node to multiple destination nodes. This is accomplished in optical networks through the technique of creating light-trees. In other words, optical signals are split into branches all-optically at the switches and crossconnects. However, the ESnet SDN core does not have the appropriate hardware for building and provisioning light-trees. Our work, therefore provides the multicast/manycast functionality as a logical overlay to the exclusively point-to-point communication paradigm possible on the network. We aim to provide point-to-multipoint by establishing a collection of point-to-point virtual circuits (VCs) and logically connecting them such that the data transfer is completed transparently to the user.

  Design & Implementation

  There are various approaches to providing the logical overlay service [5], each with their own benefits. The technique we have incorporated into OSCARS is one which establishes a collection of independent end-to-end VCs from the source to as many destinations as possible (best-effort Manycast). This option was chosen because it does not require any intermediate storage from dropping signals to the electronic layer in order to connect lightpaths originating/terminating at nodes which are not candidate destinations of the request. Our design provides mechanisms for provisioning a Multicast request as a batch of unicast requests. All multicast requests can therefore be treated individually (traditional OSCARS v0.6) and as a collective request.

  Figure 4 details and incorporation of the new multicast overlay functionality with the existing

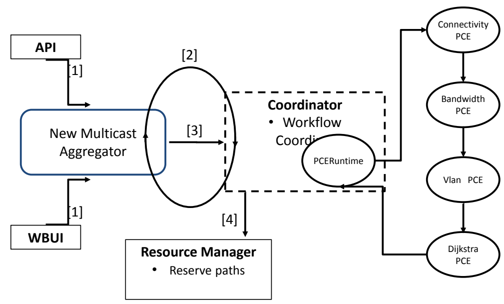

Figure 4: Point-to-multipoint overlay design.

unicast-only OSCARS system. Because the new functionality treats requests as a batch of unicast requests, the PCE stack responsible for finding an appropriate VC to satisfy each request has remained unaltered. In this way, the new manycast functionality does not impede or hinder traditional point-to-point requests from being provisioned. Further, this approach allows the user to modify one sub request in a multicast batch without needing to modify the entire batch. Tracing the above figure, the multicast/manycast behavior is as follows:

[1] User requests multicast connection in WeBUser Interface/API. The user could be a human scientist using the WBUI, or an application accessing the API. In either scenario, a request contains a single source and a group of destinations to which the user would like to provision a VC. The implementation of this multicast group is through a specialized addressing format in which all destinations may be formatted into a single destination string or listed separately in the WBUI. The multicast request is then forwarded from the WBUI/API to a logical MC Aggregator responsible for breaking down a single Multicast request into multiple unicast requests. The MC Aggregator is described as a logical component, because it is not a stand-alone entity, but is incorporated into the implementation of both the API and WBUI for simplicity. This approach allows only limited enhancement to the existing OSCARS code and prevents the need for additional protocol overheads which might be necessary if the aggregator was indeed a separate module.

[2] The MC Aggregator truncates the multicast requests into a collection of unicast requests and forwards each of these sub-requests in sequence to the Coordinator. The coordinator then forwards requests along PCE stack to identify unicast path from source to one destination, just as it would for a traditional point-to-point request. Due to the sequential nature of submitting the unicast sub-requests, a multicast request may compete with itself for network resources. Note that if a VC cannot be provisioned to one of the destinations, the multicast request is not blocked, but instead considered a manycast request in which only a subset of the specified destinations must be reached. In this way, we do not prevent partial-completion of request provisioning and assume that some sacrifice is acceptable to the user. This is in accordance with the best-effort scheme we have provided. In future extensions, we intend to provide the user an option of specifying a minimum number of destinations to be reached

before considering the request completely blocked.

[3] Each unicast sub-request from a multicast batch is handled independently and submitted to the Resource manager to reserve all nodes/ports/links in its VC. Once all sub-requests have been submitted to the coordinator, the user can then query the request to see which (if any) sub-requests could not be provisioned. The user may then look at the returned causes of failure for such sub-requests and modify the entire multicast batch, or simply modify the unicast sub-request and submit a request re-attempt with modified parameters if so desired.

Current implementation generates a unique Global Request ID (GRI) for a multicast request which contains a list of GRIs for each of its unicast sub-requests. This approach ultimately allows the user to handle sub-requests as a sub-batch of the original multicast batch. For example, if a multicast batch request aims to provision VCs to four destinations, and at some point after provisioning, the user realizes he must alter the parameters for two of the sub-requests, he can submit a modification request and pass in the GRIs of the two sub-requests and modify them together, rather than requiring two requests to perform the same operation.

- **T2.3: Survivability 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.

  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.

- **T2.4: 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 [10].

- **T3: Multi-Layer Multi-Domain Quality of Service (QoS)**

  In this task we examine the multi-layer quality of service in optical WDM networks. The aim is to deliver a QoS framework to map input connection requests to a certain number of classes, wherein each of these classes gets a different treatment in the network.

  Work Performed:

  The main tasks performed are follows: (1) development and analysis of partition-based QoS on optical WDM networks with hybrid immediate and advance reservation (IR/AR), and (2) network-wide approximate blocking analysis for hybrid IR/AR.

  As far as the first task is concerned, we followed with the design and implementation on the simulator of the partition-based QoS approach. We evaluated three network scenarios from the mixed IR/AR we investigated in Q2: (a) strict IR/AR partitioning, (b) strict IR/AR partitioning with partial sharing, and (c) flexible IR/AR partitioning with preemption. The second sub-task carried out was the development of a network-wide approximate blocking model for optical WDM networks with hybrid IR/AR. We extended the model for the link blocking analysis [11] to the whole network computation with two different WDM assumptions, under wavelength-continuity constraint and with wavelength conversion. The model is able to calculate the approximate blocking probability given a network offered load and a set of IR/AR classes. One of the contributions of the model with respect to past approaches in the literature is the addition of a flexible method to compute the blocking probability for different IR and AR traffic classes. The analytical model makes use of the Erlang fixed-point approximation to compute the network-wide blocking. As for the second sub-task, we compared the results from the analytical model with those obtained from simulation, and we can found that the blocking probability is well-approximated using the model. We tested different traffic load scenarios with different IR/AR classes and on two different network topologies, NSFnet and ESnet, showing the results that for a diverse number of wavelengths, the model can compute a good approximation of the blocking probability. This work was submitted in [12].

  We also extended our anycast algorithm to provide survivability by allowing a link-disjoint backup path to be provisioned to an alternate destination in the anycast set; this technique allows for resilient light paths for destination-agnostic applications. We compared this to the naive approach of provisioning a backup link-disjoint light path only to the primary destination in the anycast set. This approach can consume a large amount of resources because the disjoint backup path is often very long compared to the primary path. In some scenarios no disjoint path may exist to the primary destination regardless of resource availability. Our results indicate that relocation is able to significantly reduce the blocking probability compared to this naive approach because by allowing the backup path to route to an alternate destination, the probability is increased that a shorter path can be found. This reduces the overall load on the network and allows for provisioning of additional requests. Furthermore, relocation offers additional resiliency to destination failure compared to the naive approach.

NOTE: All the above work which has resulted in journal and conference papers have explicitly acknowledged the support of the DOE COMMON project.

Cost Status & Unexpected Funds

See attached document.

Next Year Deliverables

- Develop a restricted hop model for the DAMN multicast/manycast model.
- Extend survivability to multi-domain scenarios.
- For multicast OSCARS development, we plan to implement a minimum-destination parameter for canceling requests which cannot reach all specified destinations in a multicast request. Further, we also plan to enhance WBUI for easy multicast request submission.
- Extend the single domain multicast functionality to multi-domain scenarios.

References

[1] A. Gadkar, J. M. Plante, and V. M. Vokkarane, "Static multicast overlay in WDM unicast networks for large-scale scientific applications," in *Proc. ICCCN*, July 2011.

[2] A. Gadkar and J. Plante, "Dynamic Multicasting in WDM Optical Unicast Networks for Bandwidth-Intensive Applications," in *Proc. of IEEE GLOBECOM 2011*, Houston, TX, USA, Dec. 2011.

[3] A. Gadkar, J. M. Plante, and V. M. Vokkarane, "Manycasting: Energy-Efficient Multicasting in WDM Optical Unicast Networks," in *Proc. of IEEE GLOBECOM 2011*, Houston, TX, USA, Dec. 2011.

[4] J. Plante, A. Gadkar, and V. M. Vokkarane, "Dynamic manycasting in optical split-incapable WDM networks for supporting high-bandwidth applications," July. 2012.

[5] 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.

[6] 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.

[7] A. Gadkar, T. Entel, and V. M. Vokkarane, "Static Manycast Advance Reservation in Split-Incapable Optical Networks," in *Proc. of IEEE GLOBECOM 2012*, 2012, under review.

[8] T. Entel, A. Gadkar, and V. M. Vokkarane, "Dynamic Advance Reservation Multicast Overlay for Slotted Optical WDM Networks," in *Proc. of IEEE GLOBECOM 2012*, 2012, under review.

[9] M. Boddie, T. Entel, C. Guok, A. Lake, J. Plante, E. Pouyoul, B. H. Ramaprasad, B. Tierney, J. Triay, and V. M. Vokkarane, "On extending ESnet's OSCARS with a multi-domain anycast service," in *Conference on Optical Network Design and Modeling*, Essex, UK, April 2012.

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

[11] J. Triay, C. Cervelló-Pastor, and V. M. Vokkarane, "Analytical Model for Hybrid Immediate and Advance Reservation in Optical WDM Networks," in *Proc. of IEEE GLOBECOM 2011*, Houston, TX, USA, Dec. 2011.

[12] ——, "Computing approximate blocking probabilities for hybrid immediate and advance reservation in optical WDM networks," *IEEE/ACM Trans. Networking*, 2012, under revision.