# Enhancing ESnet's Unicast-Only OSCARS with a Manycast Overlay Service


Jeremy M. Plante
Department of Electrical and Computer Engineering, University of Massachusetts Lowell
1 University Ave, Lowell, MA
Jeremy_Plante@student.uml.edu

Vinod M. Vokkarane
Department of Electrical and Computer Engineering, University of Massachusetts Lowell
1 University Ave, Lowell, MA
Vinod_Vokkarane@uml.edu



## ABSTRACT

Distributed workflows are becoming the norm in scientific collaborations. Laboratories on opposite sides of the globe often exchange experimental information and transmit independent data to multiple storage or super-computing sites. Often these applications result in petabytes or exabytes of information. Such trends reveal the necessity for intelligent and efficient communication paradigms like multicasting and manycasting, which provide point-to-multipoint transfers. ESnet's *On-Demand Secure Circuits and Advance Reservations System (OSCARS)* provides logic and architectural support for directed point-to-point circuit provisioning for unicast communication on large-scale optical backbone core networks. This work aims to provide manycast support as a logical overlay to OSCARS on networks which do not have the appropriate splitting hardware at the optical layer. We implement a novel front-end manycast client that communicates directly with OSCARS. Through simulated traffic scenarios on a realistic topology, the flexibility of manycast over multicast in networks with limited splitting capabilities is demonstrated.



## Keywords

OSCARS, Manycast, VLAN, Optical


## 1. INTRODUCTION

As large-science applications expand and aim to resolve solutions and explanations to some of the world's most complex phenomena, their application workflow requirements are expanding to the point that current network architectures are becoming a bottleneck. For example, the biological and environmental research applications of the Earth System Grid Federation [1] require data replication and storage at multiple geographically dispersed data repositories. The astrophysics research conducted at the Square Kilometre Array [2] requires aggregation of information from many distributed and dissimilar instruments for combined analysis. In order to support such emerging applications, an expansion of current network services beyond the scope of traditional point-to-point communication is essential. Such applications call for sophisticated point-to-multipoint transmission protocols, like multicast and manycast, for data storage, replication, and retrieval [3].

This paper presents an overview and design of a manycast overlay for ESnet's *On-Demand Secure Circuits and Advance Reservation System (OSCARS)*, which is the network research community's most popular circuit provisioning software [4]. OSCARS provides the logic for configuring a network to establish *Virtual Circuits (VCs)* for guaranteed service on large-scale science transmissions. Recent data suggests that as much as 50% of ESnet's annual 6 petabytes of traffic is carried on OSCARS circuits [4]. At present, OSCARS supports purely point-to-point circuit provisioning. As a part of this study, a front-end OSCARS client was designed and tested to logically group individual VCs and treat them as a single manycast reservation using group operations to modify and terminate them as a unit. It should be noted that since this design was intended to not alter any existing OSCARS path computation code, the design is in keeping with the MA-VWU approach to virtual manycasting described examined in detail by the author's previous works [5, 6, 7].

Manycast is a flexible variant of multicast communication [8, 9, 10]. In multicasting, the source must communicate with all of the proposed destinations simultaneously. If even one of these destinations cannot be reached, the multicast request is deemed blocked and none of the destination nodes (including the ones that could be successfully reached) receive the transmission. In manycast however, the source is expected to only require connections to some subset of the total destination set. Destinations to reach may be selected by choosing the candidates along the cheapest paths, those destination nodes with the least load, or the most environmentally-efficient destinations [11] to provide economical resource allotment. The key difference between multicast and manycast is that in multicast, all the desired destinations are specified *a priori*, whereas in manycast the destinations must be selected (possibly dynamically) based on the state of the network. Though some multicast overlay implementations have been deployed previously, the author's proposed client is the first manycast overlay solution to be deployed on a large-scale provisioning system, and certainly the first point-to-multipoint overlay solution designed specifically for use in conjunction with OSCARS.

## 2. OSCARS OVERVIEW

Since the proposed manycast client is designed for completely front-end use with OSCARS, it is vital to have a basic understanding of the behavior of OSCARS itself in provisioning circuits. This section gives a high-level overview of the structure of OSCARS and its basic workflow for satisfying circuit requests.

OSCARS latest released version (0.6) consists of a number of service modules. Each service module is deployed as a discoverable *WebService* to be identified and queried by the other modules. The modular nature of OSCARS provides for dynamic integration of additional or customized services, straight-forward unit and integration testing, and configurable deployment, i.e. the deployer can disable certain features/services if he does not require them for his application's workflow purposes. A brief discussion of the functionality of some select OSCARS service modules follows. Though there are over a dozen of these service modules implemented in OSCARS, only those which will supply the reader with the base knowledge necessary to understand how the proposed client relies on OSCARS to perform logical manycasting are addressed.[1]

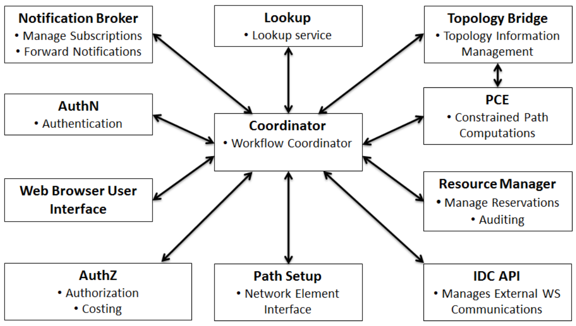



Figure 1: OSCARS modular framework.

- **Coordinator**: The Coordinator module is the central workflow engine for OSCARS. As shown in Figure 1, it is the only service module which communicates directly with all other modules. The Coordinator is responsible for passing messages from one module to another in order to establish a new, or query an existing VC.
- **PCE**: The *Path Computation Engine (PCE)* stack is arguably the single-most important service module in OSCARS, as it is the component responsible for identifying the path to reserve based on a user's input constraints (start time, end time, bandwidth, etc.). The PCE actually consists of several submodules, each of which is responsible for a different portion of the path computation. Each of these sub-modules accepts some form of user-defined constraints as well as the current network topology information and computes from this information, a modified topology which has been pruned of URNs (node/port/link combinations). The pruned URNs are those that do not meet the required criteria specified by the submodule. For example, the *BandwidthPCE* would prune out all links which do not have the available transmission rate specified by the user, while the *DijkstraPCE* would prune out all nodes and links which are not located along the shortest remaining path between the source and destination nodes in the current topology. The result of PCE stack execution is the final path which will be reserved for a given reservation, and a dedicated *Virtual LAN (VLAN)* on which that path will be established.
- **Resource Manager**: The Resource Manager is responsible for maintaining and tracking information on available network resources (bandwidth, VLANs, etc.) and must make this information available to the PCE stack during path computation to ensure proper resource-based pruning.
- **IDC API**: the *Inter-Domain Controller (IDC)* API module serves a dual purpose. It provides an interface through which other instances of OSCARS may communicate with this instance (each instance serves as an IDC for a separate topology or domain). This is useful in forwarding reservation requests and query details to networks of which the source IDC has no internal knowledge. The second purpose of this module is to provide a client API so that front-end systems may make use of OSCARS VC provisioning services while supplying additional application-specific logic. Clients will pass messages to the API (as all OSCARS modules send/receive messages) using SOAP-XML messages. This allows behavior and collaboration with OSCARS to be language independent.

[1]For greater detail on OSCARS modules, the reader is referred to [12].

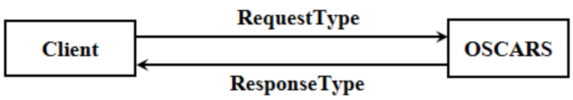



Figure 2: High-level OSCARS/client interaction.

The general behavior of a client's interaction with the IDC API is shown in Figure 2. OSCARS expects very specific types of encapsulated objects to be passed into the API, and very specific objects to be returned once it has performed its internal computations. For example, when a client application intends to reserve a new VC, it must send in a creation request object consisting of the user-constraints (bandwidth, start-time, end-time, specific VLAN, etc.) and will receive back from OSCARS a creation response object consisting of details about the reservation (success status, unique ID for the reservation, errors encountered during submission, etc.). This request/response architecture allows for client applications to create, cancel, modify, and query VC reservations using OSCARS as the underlying provisioning and management tool and adequately describes the architecture used by the proposed manycast system.

# 3. MANYCAST IMPLEMENTATION

In this section, the overlay approach used for logical manycasting in the unicast-only OSCARS environment is described. For more detailed reviews of this overlay solution including ILP formulations and comparisons to alternative overlay approaches, please refer to previous works [5, 6, 13, 14]. The overlay theory is then applied to the proposed manycast client and its behavior is detailed along with its specific interactions with OSCARS.

## 3.1 Logical Manycast Overlay Approach

For some applications, a particular sending site may require the ability to distribute data generated by various experiments to different geographical server locations across the network for independent analysis or storage. In these environments, a service provider may host a number of servers that provide the same service. For example, there may be a number of servers that can be used simultaneously for distributed data storage (and retrieval). There may also be a number of servers that can process computational tasks in parallel. The client will want to use some subset of these available resources to execute the storage or computation task. The subset in question may correspond to the lowest cost servers or servers with lowest latency. In such scenarios, manycasting can be used to select the “best" subset of servers.

Many optical networks are limited by their exclusive use of *Split-Incapable (SI)* optical switches. For example, the OXCs connecting the DOE’s ESnet are not capable of all-optically splitting an incoming signal to multiple output ports; only direct port-to-port forwarding is supported by the network hardware [15, 16]. As such, SI networks inherently fail to support the manycast communication paradigm optically. To overcome this problem, one can make use of the exclusively point-to-point connections of the optical layer and provide manycast functionality as a logical, overlay service. More specifically, for a given request, one can establish individual end-to-end lightpaths from the source node of the request to each selected candidate destination member of the request. This approach, known as *Manycasting via WDM Unicast (MA-VWU)* is the simplest form of manycast overlay support.[2] In an effort to preserve the core functionality of OSCARS, the manycast client behaves independently, thus preventing any alteration of the default PCE behavior. OSCARS calculates and provisions VCs, and the client is simply responsible for logically grouping these VCs into a single manageable unit. Despite its naïve behavior and excessive bandwidth consumption [17], MA-VWU is a suitable solution for maintaining the desired independence from OSCARS.

## 3.2 Manycast OSCARS Extension

Figure 3 gives a view of how the client manipulates OSCARS input/output to provide a logical manycast service, as well as the internal OSCARS workflow used in each reservation. As previously described, the manycast client resides in front of the IDC API as a separate service module. The client will take in user-constraints from the end-user and parse them appropriately to forward to OSCARS. In OSCARS, a reservation’s source and destination are each specified as a single URN in the network topology. The client increases flexibility in this area and allows a user to specify a *set* of URNs for the destination. The manycast client has been designed to mirror the OSCARS API so that if only a single URN is specified, the request is simply forwarded to OSCARS unmolested and the result directly returned. In other words, for a unicast VC, no alterations to the default OSCARS behavior are necessary. However, if the user specifies multiple destination URNs, each of those URNs will behave as the destination for individual OSCARS circuit requests. This is identical behavior to the MA-VWU scheme described in Section 3.1.

Each sub-reservation is handled independently, as represented by one traversal of the loop between the manycast client and the OSCARS API in Figure 3. Each of these unicast sub-reservations has its path computed by the OSCARS PCE stack, and resources provisioned by the Resource Manager. The established (or failed) reservation details will then be returned to the manycast client via the API and assigned a group ID. Once all the sub-requests have been fed into OSCARS, the end-user may perform single operations on the group with only a limited need to understand the underlying unicast VCs provided by OSCARS. The client maintains a lookup table of these group IDs, each of which maps to a set of OSCARS VC IDs. When the user performs a group operation, such as a group cancellation, the group ID is parsed into its corresponding sub-reservation IDs and the operation is passed to OSCARS as a set of independent operations on individual VCs transparent to the user, once again traversing the workflow depicted in Figure 3.

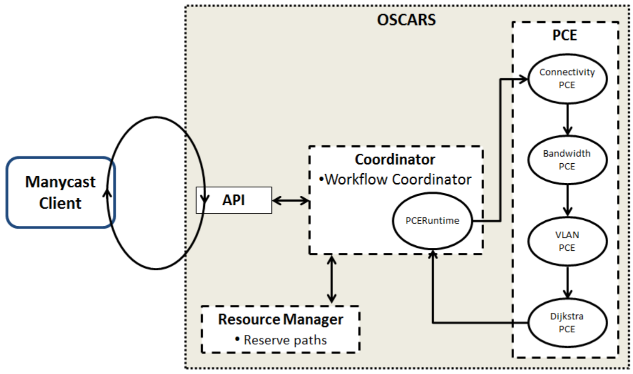



Figure 3: Manycast OSCARS client workflow.

Manycast is a very flexible communication paradigm which is actually a generalization of multicast and anycast communications.[3] The client also supports this flexibility through user-specification of additional input parameters. In addition to specifying the set of destinations to reach, the user can also specify a lower-bound (manycast threshold) and an upper-bound (manycast cutoff) indicating how many reservations to reach in order for the reservation to be deemed a success. If after all sub-requests are handled, the threshold has not been reached, then the entire reservation request is considered a failure and any successfully provisioned unicast sub-requests will be canceled *a posteriori*. For example, if the user needs to reach three destinations, but current resource availability provides capability to only reach a single destination, there is no reason to bother provisioning a circuit to that destination as the entire manycast request can no longer be satisfied. This behavior is application-specific and should be taken into account by the end-user. If the cutoff is exceeded, i.e. network resources are bountiful and more than enough destinations have been reached to satisfy the end-user’s application needs, any excess circuits will be canceled or blocked on a first-come, first-serve basis. By manipulating the threshold and cutoff values, the user may enable various communication paradigms as desired for specific applications. Table 1 gives an example of how the transmission paradigm changes with the specified values.

Table 1: Manycast OSCARS client communication paradigm

| Candidate Destinations | Manycast Threshold | Manycast Cutoff | Paradigm Description |
|---|---|---|---|
| 3 | 1 | 1 | Anycast (3/1) |
| 3 | 1 | 2 | Best-Effort Manycast (3/2) |
| 3 | 1 | 3 | Best-Effort Multicast (3/3) |
| 3 | 2 | 2 | Manycast (3/2) |
| 3 | 2 | 3 | Bounded Best-Effort Many-/Multicast |
| 3 | 3 | 3 | Multicast (3/3) |

The design of the manycast client goes beyond the scope of simply providing manycast reservation abilities. Since its virtual grouping mechanism is entirely implemented on the logical plane, this mechanism can be taken a step further. Users are not limited to perform-

[2]OSCARS does not use wavelength-routed networks, however the behavior of MA-VWU is easily applied to VCs.

[3]Anycast refers to a service request with a single source and a destination set, from which a single destination is selected to satisfy the request. The destination may be selected using the same approaches described for manycast.

ing group operations on those sub-requests which were submitted to OSCARS together. Flexibility has been provided to create dynamic groups of reservations consisting not only of unicast VCs, but also of one or more sub-groups. In this manner, if a user has previously specified a number of manycast reservations which are no longer needed, he can add them to one super-group and cancel the entire group at once rather than canceling each of them independently. This may also be particularly useful if the user's constraints change after reservations have been created. The user can group the reservations into a unit and perform some update on all of them at once. This dynamic grouping capability allows the end-user the ability to categorize certain sub-groups or sub-reservations into distinct classifications according to the application's requirements. Members can also be removed from groups dynamically or treated as independent entities of their parent groups. For example, suppose a user issues a manycast reservation that results in three OSCARS VCs being successfully provisioned. The manycast client does not mandate that all three of those VCs must be operated on simultaneously. The user is still able to perform independent functions on sub-requests to enhance flexibility of use.

## 4. QUANTITATIVE CLIENT EVALUATION

The proposed manycast OSCARS client has been subjected to various dynamic traffic scenarios to examine the behavior of both multicast and manycast request sets on the ESnet topology shown in Figure 4. Note that this particular version of ESnet contains several instances of dual-fiber links between nodes. This topology is representative of a realistic network for which OSCARS might provide its provisioning services and provides challenges to resource provisioning not considered by previous theoretical results. All links (single-fiber) in the ESnet topology are bidirectional and are assumed to have 10 Gbps bandwidth capacity. For each arriving manycast request, the source node and manycast destination nodes are uniformly distributed, while the request's bandwidth demands are uniformly distributed in the range [1 Gbps, 5 Gbps], in increments of 1 Gbps. Note that this bandwidth demand is the transmission rate which must be guaranteed to *each* destination reached. For example, a manycast reservation that must provision VCs to two destinations with guaranteed bandwidth of 5 Gbps would fill any overlapping links in their respective paths to capacity. As a result, blocking rates will be very high. As previously discussed, reservations are blocked only if the manycast threshold cannot be realized. A sufficient quantity of VLANs is supported to prevent a resource shortage, thus the only resources which requests compete for are time and bandwidth.

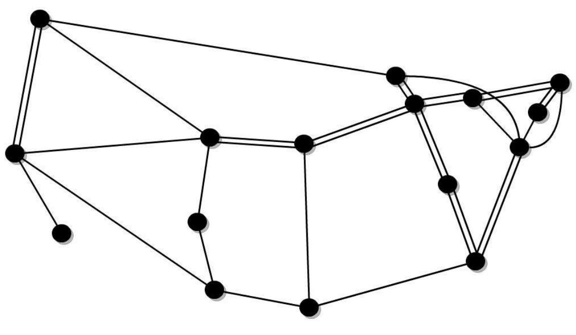


Figure 4: 16-node ESnet science data core network used in evaluating advance reservation traffic scenarios using the manycast OSCARS client.

Unlike the immediate reservation simulations conducted to yield the results presented in [6, 7, 13], all of the simulated OSCARS requests here are *Advance Reservations (AR)*, which arrive to the system some time before they need to be provisioned, reserve resources for a fixed duration at some specified time in the future, and then depart the system and free their dedicated resources immediately upon termination of that duration. All requests are scheduled to reserve, transmit, and release network resources within a two-hour time window. A request set's *correlation factor* corresponds to the probability that requests overlap during that time window. As the correlation factor increases, more requests overlap in time; a correlation factor of zero provides a set of completely time-independent reservations which do not compete amongst each other for network resources. The formula for calculating the correlation factor for a set of requests is given as $\sum_j C_j/n(n-1)$, where $n$ is the number of requests to schedule, and $C_j$ is the number of requests which overlap in time with request $j$ [18]. Please note that the correlation factor does not directly represent load on the network, as the overlapping requests are in fact manycast requests targeting multiple destinations at any given point in time. All results shown in this section represent the average of 30 unique sets of reservation requests, with each set consisting of 100 manycast requests.[4]

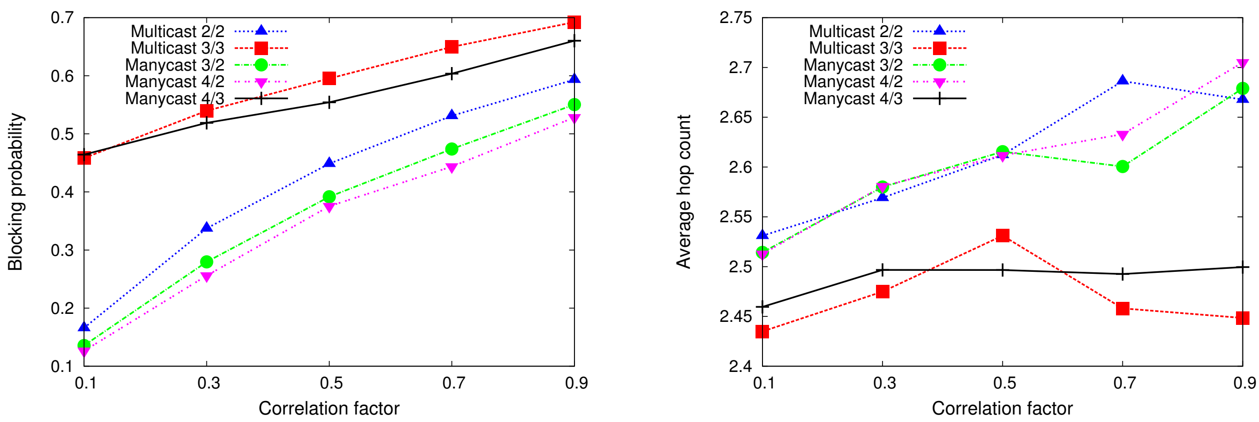
Figure 5(a) depicts the blocking probability of manycast OSCARS reservations under various traffic scenarios. By manipulating the threshold and cutoff values, it is possible to simulate both multicast and manycast reservations. It can be observed from the figure that in all cases, the additional flexibility provided by manycast destination-selection reduces reservation blocking despite needing to reach the same number of destinations as its multicast counterpart. Note that in the cases where two destinations must be reached by every request, adding a third manycast candidate destination noticeably reduces request blocking, and that adding a fourth reduces blocking even further.

Figure 5(b) shows the corresponding average hop count for successfully provisioned OSCARS VCs. All scenarios result in a hop count between 2.4 and 2.75 hops. Both the multicast and manycast scenarios reaching three destinations have fewer hops than their two-destination counterparts. This is related to their corresponding blocking rates shown in Figure 5(a); a greater number of destinations yields greater resource consumption and thus greater overall blocking. As these resources are consumed, fewer long paths are able to be provisioned and thus only short paths will be successfully reserved, thereby lowering the average hop count. It is worth noting that the multicast $2/2$ hop average is greater than both the manycast $3/2$ and manycast $4/2$ averages at a correlation factor of $0.7$. It is relatively equal to its counterparts for most other correlation factors, but at this particular factor, not only is the blocking greater, but the hop count is longer too. This data point demonstrates the relative inefficiency of multicast when compared to manycast in certain configurations.

## 5. CURRENT LIMITATIONS AND FUTURE ENHANCEMENTS

As discussed in previous sections, the fact that the manycast OSCARS client is a practical implementation of the MA-VWU overlay model inherently means that the solution will minimize the number of logical hops, and ultimately delay across the circuit. It

[4] MA-VWU has been previously assessed theoretically for different network topologies under static AR traffic scenarios [5].




(a) Blocking probability.

(b) Hop count.

Figure 5: Comparison of multicast and manycast OSCARS client.

also eliminates the need for identifying network nodes capable of performing internal storage or data buffering before forwarding the data to the next hop in the overlay tree as in the multiple-hop MA-DMN and MA-DAN models explored in [5]. This is particularly a benefit when considering that OSCARS VCs are used to provide guaranteed service to transmit huge quantities of experimental data which are often destined to super-computing facilities, as these sites are likely the only ones with enough storage and computational capacity to process the data. However, since the manycast client is an exclusively front-end solution to the overlay problem and does not manipulate the logic that OSCARS uses to provision its underlying circuits, it is not an ideal solution in some respects.

As discussed previously, MA-VWU leads to a large overhead in resource consumption. This is a serious problem in OSCARS, which supports only layer-2 circuit provisioning with no assumed storage capabilities at the ingress access nodes. This means that the source for any reservation is a single port on a particular network node. Consider the illustrative example shown in Figure 6, wherein a user submits a manycast reservation through the client to reach three destinations across the network. The source for the reservation is the node-port combination $N_1$:$P_1$, while the shortest paths to the three destinations will traverse $N_1$:$P_2$, $N_1$:$P_3$, and $N_1$:$P_4$ respectively. For the sake of argument, let the user's requested bandwidth be specified as 3 Gbps, while the total link capacity anywhere in the network is 9 Gbps. Considering layer-3 provisioning, a unique circuit would be provisioned to each destination, consuming 3 Gbps along each intermediate link to the destinations, leaving plenty of room for future reservations sourced at, or traversing $N_1$. However, when considering layer-2 provisioning, it becomes obvious that the MA-VWU implementation consumes 3 Gbps on the source port for *each* of the three VCs; a total of 9 Gbps for the entire manycast reservation. This means that future reservations which need to traverse $N_1$:$P_1$ will be blocked due to the resulting bandwidth-resource shortage. In general, given any manycast reservation destined to $K'$ destinations, each sub-reservation may only reserve at most $1/K'$ of the total available bandwidth capacity of the source port's associated link. The proposed manycast overlay client may be extended to be compatible with a VPLS network framework, which will enable the minimization of port-consumption for each manycast session [19, 20].

Another shortcoming of the MA-VWU implementation in the scope of OSCARS is that every circuit is established along a separate VC. This means for a manycast reservation destined to $K'$ nodes, there will be exactly $K'$ VCs provisioned in the network by OSCARS. Therefore, should a particular end-user want to transmit data to all the destinations, the manycast client is required to supply a VLAN lookup table to determine what VLANs are accessible. This of course limits the transparency from the perspective of the user who ultimately still needs knowledge about the individual unicast VCs that constitute the greater manycast reservation. A manycast reservation consisting of multiple logical hops (rather than a single hop from the source for each circuit) would not necessarily suffer from this issue as it is possible to use the same VLAN (assuming it is available) along each logical hop in an overlay tree, much in the same way it can provision the same wavelength across the entire tree in the illustrative example offered in [5]. This is especially true when assuming that the drop-nodes have replication capabilities; any shared links among paths to distinct destinations would need to carry only one copy of the transmitted data and only replicate it at diverging drop-points.

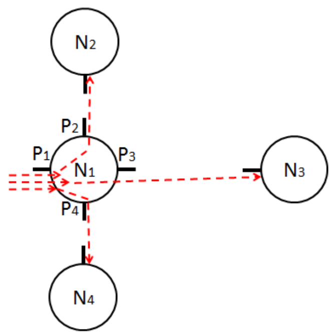



Figure 6: Illustration of inefficient MA-VWU OSCARS client source port bandwidth consumption for a simple 3-destination manycast reservation.

Future improvements to the existing manycast client application include the consideration of OSCARS overlay solutions incorporating the more complex MA-DMN/MA-DAN approaches and evaluating trade-offs with the current MA-VWU implementation in a practical system. It is also worth exploring alternative protocols for canceling unneeded reservations when the manycast cutoff of a reservation is exceeded. Currently, the first successful unicast VCs will remain provisioned, while those VCs provisioned latest will be canceled. It might be desirable to extend the client's cancellation functionality to cancel the longer paths, thereby reducing network resource consumption for each group reservation. Alternatively, the end-user may be provided with the ability to prioritize specific destinations, and allow the client to cancel VCs to the manycast candidate destinations not in that specified group in the event that the manycast cutoff is exceeded.

Despite the need for extension, the proposed manycast client presents a novel approach to manycasting in SI networks. From a network perspective, the proposed client is not as efficient as it could be, however, from an end-user point of view it allows manycast functionality where previously none was supported. The first large-scale deployable manycast service for optical core networks has been successfully developed by incorporating the proposed manycast client with OSCARS.

## 6. ACKNOWLEDGMENTS

This work has been supported by the DOE COMMON project under grant DE-SC0004909.

## 7. REFERENCES

[1] "Earth System Grid Federation," 2013, [Online]. Available: http://esg-pcmdi.llnl.gov/esgf.

[2] "Square Kilometre Array," 2013, [Online]. Available: http://www.skatelescope.org/.

[3] C. Guok and T. Lehman, "Intelligent network services: A vision for DOE network services evolution," Aug. 2012, [DRAFT]. Available: http://archstone.east.isi.edu/twiki/pub/ARCHSTONE/Documents/intelligent-network-services-v1.2.pdf.

[4] "Virtual circuits (OSCARS)," 2013, [Online]. Available: http://www.es.net/services/virtual-circuits-oscars/.

[5] T. Entel, A. G. Gadkar, and V. M. Vokkarane, "Static manycast advance reservation in split-incapable optical networks," in *Proceedings, IEEE International Conference on Communications (ICC)*, Budapest, Hungary, Jun. 2013.

[6] A. Gadkar, J. Plante, and V. M. Vokkarane, "Manycasting: Energy-efficient multicasting in WDM optical unicast networks," in *Proceedings, IEEE Globecom*, Houston, TX, Jul. 2011.

[7] J. Plante, A. Gadkar, and V. M. Vokkarane, "Dynamic manycasting in optical split-incapable WDM networks for supporting high-bandwidth applications," in *Proceedings, International Conference on Computing, Networking and Communications (ICNC)*, Maui, HI, Feb. 2012.

[8] S. Y. Cheung and A. Kumar, "Efficient quorumcast routing algorithms," in *Proceedings, Infocom*, Toronto, Ontario, Canada, June 1994, pp. 840–847.

[9] R. Ravi, R. Sundaram, M. V. Marathe, D. J. Rosenkrantz, and S. Ravi, "Spanning trees short or small," in *Proceedings, ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Arlington, VA, 1994, pp. 546–555.

[10] B. G. Bathula and V. M. Vokkarane, "QoS-based manycasting over optical burst-switched (OBS) networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 271–283, 2010.

[11] T. Schöndienst, J. M. Plante, D. A. Davis, and V. M. Vokkarane, "Energy source-aware manycast overlay in WDM networks," in *Proceedings, IEEE Globecom*, Atlanta, GA, December.

[12] ESnet OSCARS Development Team, "Oscars," 2012, [Online]. Available: http://oscars-idc.googlecode.com/svn-history/r613/wiki/OSCARS06Architecture/oscars0_6-architecture-2012.pptx.

[13] A. Gadkar, J. Plante, and V. M. Vokkarane, "Multicast overlay for high-bandwidth applications over optical WDM networks," *Journal of Optical Communications and Networking*, vol. 4, no. 8, pp. 571–585, Aug. 2012.

[14] T. Entel, A. Gadkar, and V. M. Vokkarane, "Scheduled multicast overlay for bandwidth-intensive applications," in *Proceedings, International Conference on Optical Networking Design and Modeling (ONDM)*, Colchester, UK, 2012, pp. 1–6.

[15] "Networking for the future of DOE science," 2008, [Online]. Available: http://www.es.net/assets/Uploads/ESnet4-Networking-for-the-Future-of-Science-2008-05-05.NP.v1.pdf.

[16] C. Guok, D. Robertson, E. Chaniotakisy, M. Thompson, W. Johnston, and B. Tierney, "A user driven dynamic circuit network implementation," in *IEEE GLOBECOM Workshops*, 2008, pp. 1–5.

[17] T. Szymanski, "Future internet video multicasting with essentially perfect resource utilization and QoS guarantees," in *Proceedings, IEEE International Workshop on Quality of Service*, San Jose, CA, 2011, pp. 1–3.

[18] N. Charbonneau and V. Vokkarane, "Static routing and wavelength assignment for multicast advance reservation in all-optical wavelength-routed WDM networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 1–14, 2012.

[19] K. Kompella and Y. Rekhter, "RFC 4761: Virtual Private LAN Service (VPLS) using BGP for auto-discovery and signaling," Jan. 2007.

[20] M. Lasserre and V. Kompella, "RFC 4762: Virtual Private LAN Service (VPLS) using Label Distribution Protocol (LDP) signaling," Jan. 2007.