Static Manycast Routing and Wavelength Assignment over Wavelength-Routed Optical WDM Networks

Neal Charbonneau and Vinod M. Vokkarane
Department of Computer and Information Science, University of Massachusetts, Dartmouth, MA
E-mail: u_ncharbonne@umassd.edu and vvokkarane@umassd.edu

***Abstract*—We present our initial work for static manycast routing and wavelength assignment (MA-RWA) over wavelength-routed optical WDM networks. The goal is to route a set of static manycast requests over a wavelength-routed WDM network while minimizing the number of wavelengths required. This is the first time the problem has been investigated. We present a lambda path heuristic (LPH) to solve the MA-RWA problem and compare it to a simple shortest path manycast heuristic. LPH achieves a 20-30% improvement in required wavelengths over realistic networks.**

Keywords: WDM, wavelength-routing, manycast, and RWA.

I. Introduction

Future Internet applications, such as IPTV, cloud storage/computation, video conferencing, and peer-to-peer (P2P), will require large amounts of bandwidth and support for point-to-multipoint communication. To support these applications, the next-generation Internet will be based on optical networks that can provide huge amounts of bandwidth. Manycast [1], [2] is a communication paradigm that can support the point-to-multipoint nature of future applications, in addition to supporting traditional communication paradigms. Manycast supports communication from a sender to any k out of m ($k \leq m$) candidate destinations where the candidate destination set, $|D_c| = m$, is a subset of nodes in the network. If we change the parameters of the manycast request, we can also perform unicast ($k = m = 1$), multicast ($k = m > 1$) and anycast ($k = 1 < m$). Manycast is a powerful communication framework that is important for next-generation applications [3]. Since the future Internet will based on optical networks, it is important to support manycast over wavelength-routed networks.

In this work we will consider the static MA-RWA problem. In this problem we are given a set of manycast requests and for each request we must assign a route tree (or light-tree [4]) and a wavelength. The goal is to minimize the number of wavelengths required. We can define a manycast request as (s, D_c, k) where s is the source, D_c is the candidate destination set, and k is the number of nodes necessary to reach out of D_c. This is related to the multicast problem, but is more general. In multicast, the destinations are specified ahead of time, in manycast the destinations must be chosen. To solve the multicast problem, a Steiner tree must be generated, which has been shown to be NP-hard [2]. Since the destinations must be chosen in manycast, there are $\binom{|D_c|}{k}$ combinations of nodes to use in the creation of a Steiner tree. Since manycast is a generalization of multicast, it is also NP-hard.

Supporting manycast over optical networks is important because supporting point-to-multipoint communication with unicast results in wasted resources at the optical layer [5]. Solving MA-RWA will help in dimensioning and enabling future networks to support new applications. Consider grid computation networks having replicated services. MA-RWA can choose the appropriate destinations for long term requests that are known in advance. Also consider content retrieval from replicated storage for large-scale e-Science applications. Given a set of nodes that store data, MA-RWA can choose which nodes to select the data from out of the candidate set.

The paper is organized as follows. Section II will discuss related work, Section III will discuss our assumptions, problem definition, and proposed heuristic, Section IV will provide numerical results, and Section V concludes the paper.

II. Related Work

Quorumcast, which is a specific case of manycast where $k = \left\lceil \frac{|D_c|}{2} \right\rceil$, was proposed by [1], [2]. Since then, a number of quorumcast routing algorithms have been proposed [1], [6], [7], [8]. Manycast has also been proposed over optical burst-switched networks [9], [10], [11], [12]. The main challenge for manycast over OBS is providing reliability despite random contentions. These works focus on dynamic traffic and distributed routing algorithms or unicast routing algorithms to provide reliable manycast for OBS. These approaches typically do not setup a route tree for each request. The authors in [13] propose an ILP and several heuristics for solving multi-resource manycast in mesh networks. Recently, an anycast RWA algorithm was proposed for wavelength-routed networks [14]. Anycast is a specific instance of manycast.

To the best of our knowledge, this is the first time static MA-RWA has been proposed for wavelength-routed networks. The contributions of this work include being the first to address the static manycast RWA problem for wavelength-routed networks and the development of a heuristic to solve it. It is also shown that simply turning a manycast request into a multicast request at the source is not an efficient way to support manycast.

III. Static Manycast RWA: Problem Statement and Proposed Heuristic

A. Problem Definition and Assumptions

The static MA-RWA problem is defined as follows. Given a network $G = (V, E)$ and a set of manycast requests $M = \{(s_1, D_{1,c}, k_1), (s_2, D_{2,c}, k_2), ..., (s_n, D_{n,c}, k_n)\}$ where $s_i \in V$, $D_{i,c} \subset V$, and $k_i \leq |D_{i,c}|$, we must find a route tree, or light-tree, and wavelength assignment for each manycast request in M such that the number of wavelengths required is minimized. Since the requests are static, this is done offline. We use a single route tree for each request and assume one wavelength per request. We assume wavelength converters are not available so each tree must satisfy the wavelength continuity constraint. In other words, each tree must use the same wavelength on all links. Also, no two light-trees can use the same wavelength over the same link.

Algorithm 1 Lambda Path Heuristic for static MA-RWA.

1: $sort\_desc(M)$
2: **for all** $m\ in\ M$ **do**
3: $D = \{\}$
4: $allTrees = list()$
5: **while** $D_{m,c} - D \neq \phi$ **do**
6: $T = (V', E')\ s.t.\ V' = \{s_m\}, E' = \phi$
7: $path = min\{SP(s_m, u)\}\ u \in D_{m,c} - D$
8: $Update(T, path)$
9: $D = \cup\{u\}$
10: $copy = 1$
11: **while** $copy < k$ **do**
12: $path = min\{SP(u_1, u_2)\}\ u_1 \in V', u_2 \in D_{m,c} - V'$
13: $V' = V' \cup \{u_2\}$
14: $Update(T, path)$
15: $copy = copy + 1$
16: **end while**
17: $T.cost = \sum_{i,j \in E'} c_{i,j}$
18: $T.newWL = increasesWL(G, T)$
19: $allTrees.append(T)$
20: **end while**
21: $T = min(allTrees)$
22: $FirstFit(G, T)$
23: **end for**

We assume all nodes in the network are able to split an incoming signal to any number of output ports. These types of switches are known as multicast-capable optical cross connects (MC-OXCs). We also do not consider impairment or power-awareness in this paper. In a realistic scenarios, especially with splitters, the power and signal-to-noise ratio should be taken into account for routing in optical networks, but we will investigate this in future work.

B. Lambda Path MA-RWA Heuristic

In this section we propose a heuristic for static MA-RWA. The heuristic orders the static request set and satisfies each request individually trying to minimize the number of wavelengths required. We order the requests in descending order of k_i so requests with larger number of required destinations are satisfied first. To satisfy each individual request, we use a modified version of the improved path heuristic (IMP) [1]. We modify it to include wavelength assignment, an additional constraint to minimize wavelengths required, more cost functions, and to also iterate over more Steiner trees. We will refer to the new heuristic as lambda path heuristic (LPH).

The heuristic is shown in Algorithm 1. Before describing how it works, we will define the functions that it uses. First, the SP function finds the shortest path between the two nodes specified as parameters. The $Update$ function adds a path to the specified tree T. The $increasesWL$ function determines if assigning a wavelength to a tree, using First-Fit, would require an increase in the wavelength count given the wavelengths currently used for the previous requests. Lastly, the $FirstFit$ function performs First Fit wavelength assignment given the generated tree and current state of the network.

To satisfy a manycast request, there are $\binom{|D_c|}{k}$ possible combinations of nodes that can be used to create Steiner trees. This heuristic works by creating just $|D_c|$ Steiner trees. Each tree is generated using the shortest-path heuristic proposed in [15]. $|D_c|$ Steiner trees are created by forcing selection of the first node in line 7 when building the tree. During the first iteration, the shortest-path node is selected, in the next iteration the next shortest-path node is selected, and so on. Each time a node is selected to be the initial node it is added to D. Iteration terminates when $D = D_c$. This ensures that all nodes in D_c are in at least one Steiner tree and that multiple trees are generated. The goal of this is to find a good Steiner tree without having to try all $\binom{|D_c|}{k}$ combinations of nodes. After the generation of each tree, line 18 checks to see if assigning a wavelength according to First-Fit would result in an increase in wavelength count required. Once the trees have been generated, the minimum cost tree that requires no increase in wavelength count using First-Fit is chosen. If there is no such tree, just the minimum cost tree is chosen. We consider different cost functions described in Section IV.

Assuming that the shortest paths are pre-computed ($O(n^3)$), the heuristic runs in $O(k|D_c|^2)$ time for each individual request. The heuristic is run once for each of the n requests.

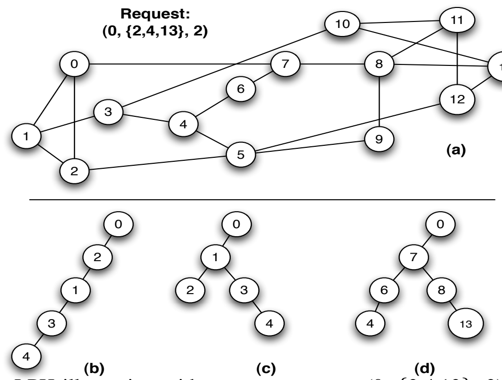

Fig. 1. LPH illustration with manycast request (0, {2,4,13}, 2). Given the network and request in (a), LPH generates the Steiner trees shown in (b-d).

C. Illustration

We will now provide an example of the LPH heuristic for a single manycast request. The request and network (NSFnet) are given in Fig. 1(a). The request, $(0, \{2, 4, 13\}, 2)$, means that node 0 is the source, nodes 2, 4, and 13 are the candidate destinations, and two out of the three destinations must be reached. LPH will iterate three times ($|D_c|$), each time choosing a different starting node to form a tree. On the first iteration, node 2 is chosen since it is the shortest path distance from 0 $(0 \rightarrow 2)$. Node 4 is added by concatenating $(2 \rightarrow 1 \rightarrow 3 \rightarrow 4)$ to the tree. This results in the tree seen in (b) which covers nodes 2 and 4. In the next iteration, node 4 is chosen to start the tree $(0 \rightarrow 1 \rightarrow 3 \rightarrow 4)$. A new branch can then be added to a create tree reaching node 2 $(1 \rightarrow 2)$, as seen in (c). Lastly, node 13 is chosen first with path $(0 \rightarrow 7 \rightarrow 8 \rightarrow 13)$. The tree can then be modified to branch at node 7, reaching node 4 $(7 \rightarrow 6 \rightarrow 4)$, as seen in (d). The iterations are now complete since every node in D_c has been used as a starting node. The heuristic is now able to choose the best tree. It will choose the lowest cost tree that does not require an increment in the number of wavelengths used in the network. If this is not available, the lowest cost tree will be chosen. The same heuristic is run for all requests.

IV. Numerical Results

To evaluate our proposed heuristic, we use it for static MA-RWA over four different networks. We used a scaled

TABLE I

COMPARISON OF LPH AND SPT OVER DIFFERENT NETWORK TOPOLOGIES FOR A STATIC SET OF 150 MANYCAST REQUESTS.

Netw.	V	E	δ	τ	$D_c^m = 10$						$D_c^m = 8$						$D_c^m = 6$					
					LPH-S		LPH-D		SPT		LPH-S		LPH-D		SPT		LPH-S		LPH-D		SPT	
					w_a	d_a	w_a	d_a	w_a	d_a	w_a	d_a	w_a	d_a	w_a	d_a	w_a	d_a	w_a	d_a	w_a	d_a
ATT	27	41	3.0	3.4	48.1	5.4	52.2	5.2	63.1	5.3	45.9	5.4	48.9	5.2	59.7	5.3	42.6	5.5	45.8	5.2	55	5.4
NSFnet	14	21	3	4.3	51.4	8.2	52.2	7.8	75.5	8.0	46.2	8.4	51.3	7.9	68.2	8.0	44.1	8.4	47.8	7.9	64	8.2
Italy	21	36	3.4	0.6	56	1.6	58.1	1.4	71.4	1.5	51.6	1.5	54.9	1.4	67.1	1.5	48.1	1.5	51.5	1.5	61.7	1.5
24-node	24	43	3.6	3.9	36.4	10.7	39	10.1	53.4	10.3	34.4	10.6	37.3	10.2	51.4	10.4	32.9	10.7	35.2	10.2	48.6	10.4

version of the AT&T network, NSFnet, the Italian WDM network, and a 24-node mesh.We use the link distances for shortest path routing and to calculate delay. We generate a set of 150 requests for static MA-RWA. The source node for each request is uniformly distributed over all nodes. For each request, the size of $D_{i,c}$ is uniformly distributed from $3..D_c^m$ and $k = \left\lceil \frac{D_{i,c}}{2} \right\rceil$. The destination nodes are also uniformly distributed across the network for each request. We calculated 95% confidence intervals for all results but do not include them here because they are negligible.

To evaluate the effectiveness of our proposed heuristic, we compare it to a shortest path heuristic, called shortest path tree (SPT). For each request, SPT chooses k out of D_c that are closest to the source according to the shortest paths. It then uses the minimum path heuristic (MPH) [15] to create a Steiner tree to these destinations. SPT essentially treats the request as a multicast request since it only considers the closest k nodes instead of all possible nodes. The main difference between SPT and LPH is that LPH considers *multiple* Steiner trees by including different nodes while SPT makes a *single* decision on which nodes to include. MPH is a 2-approximation for Steiner trees, so we use it as a bound to show that improvements can be made by taking into account the ability to choose destinations dynamically instead of simply choosing k shortest path nodes for the manycast request.

We mentioned in Section III-B that the minimum cost tree is chosen in the LPH heuristic. We define two cost functions to choose a tree. The first, called *size* (LPH-S), chooses a tree that uses the least number of links, or the smallest tree, to satisfy the request. The reasoning behind this cost function is that smaller trees will leave more resources available for more requests. The other is *delay* (LPH-D), which chooses the tree that has the lowest delay, which is the average delay over all source-destination pairs on the tree.

In Table I we compare SPT, LPH-S, and LPH-D. The network characteristics are given in the table where V is the number of nodes, E is number of edges, δ is average nodal degree, and τ is average delay per edge (ms). We ran each heuristic with different maximum destination set sizes, D_c^m, and recorded the average number of wavelengths required, w_a, and average tree delay, d_a (ms), for 150 requests.

The results show that for the number of wavelengths required (w_a columns), there is a 20-30% improvement using LPH-S compared to SPT, and LPH-S performs slightly better than LPH-D. For example, with $D_c^m = 10$ over NSFnet we can see that LPH-S requires on average 51.4 wavelengths while SPT required 75.5 for 150 requests. The AT&T and Italian networks see an average of 22% improvement while the NSFnet and 24-node network seen an average of 31% improvement. As expected, as the maximum candidate set size decreases, the number of wavelengths required decreases.

Many next generation applications require low delay, so it is important that LPH does not create trees with much higher delay. The table also shows that the differences in delay (d_a) are not significant. As expected, using the delay cost metric (LPH-D) provides the lowest average tree delay. However, the differences across the different algorithms are less than $1ms$. Due to this, we conclude LPH-S performs the best.

We ran simulations for manycast request set sizes between 50 and 200 and found the results are similar in all cases except for the Italian WDM network. There the performance improvement drops to about 15% when the request size is 50.

V. CONCLUSION

In this paper we presented, for the first time, an efficient heuristic for static MA-RWA. We compared the LPH heuristic to a simple shortest path tree heuristic and showed a 20-30% improvement in the number of wavelengths required.

REFERENCES

[1] S. Y. C. et. al., "Efficient quorumcast routing algorithms," in *Proceedings, IEEE INFOCOM*, June 1994, pp. 840–847.

[2] R. R. et. al., "Spanning trees short or small," in *Proceedings, ACM-SIAM symposium on Discrete algorithms*, 1994, pp. 546–555.

[3] R. Jain, "Internet 3.0: Ten problems with current Internet architecture and solutions for the next generation," in *Proceedings, IEEE MILCOMM*, Oct. 2006.

[4] L. H. S. et. al., "Light trees: optical multicasting for improved performance in wavelength routed networks," *IEEE Communications Magazine*, vol. 37, no. 2, pp. 67–73, Feb 1999.

[5] R. M. et. al., "Benefits of multicasting in all-optical networks," in *SPIE Proceedings, All-Optical Networking*, 1998, pp. 209–220.

[6] B. D. et. al., "Quorumcast routing by multispace search," in *Proceedings, IEEE GLOBECOM*, vol. 2, Nov. 1996, pp. 1069–1073.

[7] C. P. Low, "Optimal quorumcast routing," in *Proceedings, IEEE GLOBECOM*, vol. 5, Nov. 1998, pp. 3013–3016.

[8] B. Wang and J. C. Hou, "An efficient QoS routing algorithm for quorumcast communication," in *Proceedings, IEEE ICNP*, 2001.

[9] X. H. et. al., "Manycasting over optical burst-switched networks," in *Proceedings, IEEE ICC*, Jun. 2007.

[10] Q. S. et. al., "Multi-resource manycast over optical burst-switched networks," in *Proceedings, IEEE ICCCN*, Aug. 2007, pp. 222–227.

[11] B. G. B. et. al., "Impairment-aware manycast algorithms over optical burst-switched networks," in *Proceedings, IEEE ICCCN*, 2008.

[12] ——, "QoS-based manycasting over optical burst-switched (OBS) networks," *IEEE/ACM ToN*, May 2009.

[13] Q. S. et. al., "On finding minimum cost tree for multi-resource manycast in mesh networks," *OSN*, vol. 6, 2009.

[14] D. Din, "A hybrid method for solving ARWA problem on WDM network," *Computer Communications*, vol. 30, no. 2, pp. 385–395, 2007.

[15] H. Takahashi and A. Matsuyama, "An approximate solution for the steiner problem in graphs," *Math. Japonica*, 1980.