Routing and Wavelength Assignment of Static Manycast Demands Over	Định tuyến và gán bước sóng cho các yêu cầu Manycast tĩnh trên các mạng

All-Optical Wavelength-Routed WDM Networks	WDM định tuyến bước sóng toàn quang
Abstract	Tóm tắt
In this paper we present the static many-cast routing and wavelength assignment (MA-RWA). Manycast is a point-to- multipoint communication paradigm with applica-tions in e-Science, Grid, and cloud computing. To solve MA- RWA, a light-tree must be assigned to each manycast request in a static set such that the number of wave-lengths required is minimized.	Trong bài báo này, chúng tôi trình bày kỹ thuật định tuyến và gán bước sóng many-cast (MA-RWA). Manycast là mô hình truyền thông từ một điểm đến nhiều điểm có nhiều ứng dụng trong e-Science, Điện Toán Lưới, và Điện Toán Đám Mây. Để giải quyết vấn đề MA- RWA, cần phải ấn định một cây quang (đường quang, tuyến quang) cho mỗi yêu cầu manycast trong một bộ tĩnh sao cho số bước sóng cần thiết tối thiểu.
I. INTRODUCTION	[redacted]
The combination of a route and wavelength is known as a lightpath [3]. A static traffic model gives all the traffic demands between source and destinations ahead of time. Traditionally, communication in a network is unicast, where a single source sends data to a single destination. In this work, we consider a communication paradigm called manycast [5-7]. We can define a manycast request as a three-tuple, (s,D_c,k), where $s \in V$ is the source, Dc $\subseteq$ V is the candidate destination set, and $k \leq \|Dc\|$ is the number of nodes necessary to reach out of Dc. This means that the source node will send data, simultaneously, to some subset of size k of the candidate destination set. This is a generalization of the multicast communication paradigm. In the multicast problem, the source communicates with all of D,	

simultaneously. Given our definition of manycast, if we have k = |Dc|, then the manycast request becomes a multicast request. Since we can define multicast as a specific instance of manycast, we consider manycast a generalization of multicast.

Manycast and multicast are related in that they both require setting up a tree in the network instead of a path so that the source can communicate with multiple destinations. To efficiently support manycast or multicast requests, the network must create light-trees [8]. The problem of finding the optimal route for a light- tree is equivalent to finding the minimal Steiner tree, which is known to be NP-complete [9], although efficient approximations exist. These switches are known as multicast-capable OXCs (MC-OXCs). Multicast-capable reconfigurable optical add-drop multiplexers (ROADMs) may also be used [12]. The key difference between multicast and manycast is that in multicast, the destinations are specified ahead of time, whereas in manycast the destinations must be chosen (based on the state of the network, for example). To solve the multicast problem, we have a single set of nodes that are used to find the optimal Steiner tree connecting the source and destination set. In manycast, we must choose a subset of k nodes; this means that there are a total of $\binom{|D_c|}{k}$ combinations of nodes to use in the creation of a Steiner tree. The manycast problem, though similar to multicast,

[redacted]

[redacted]

requires a new set of solution techniques.

In this work we will consider the static manycast routing and wavelength assignment (MA-RWA) problem. In this problem we are given a set of manycast requests and for each request we must assign a light- tree. The objective is to minimize the number of wavelengths required to satisfy all the manycast requests.

We can use manycast to choose some subset of these locations (e.g., the lowest-cost storage clusters) to send this data in parallel along a light-tree set up by the network. From the network operator perspective, manycast would allow the network to optimize its resources, for example, by load balancing. Manycast allows the freedom to choose different nodes depending on the state of the network, leading to better performance for user applications and better utilization for the network. Providing a manycast service is to provide many- cast at the application layer supported by unicast at the optical layer. The authors in [15] show the benefit of supporting multicast directly at the optical layer instead of higher layers (IP in this case).

This is the first paper, to our knowledge, that investigates manycast over wavelength-routed networks. Manycast is also known as quorumcast or the k-Steiner problem. It was proposed in [5,6]. Since then, a number of quorumcast routing algorithms have been proposed [5,25–27]. Finding a

[redacted]

minimumcost tree for a manycast request is NP-hard [28]. Manycast is also related to the k-MST problem [28] in that a ρ-approximation algorithm for k-MST leads to a 2-ρ approximation algorithm for manycast. In this work we have a set of static requests and we must also perform routing and wavelength assignment. Manycast has also been proposed over optical burstswitched (OBS) networks [7,29–31] In manycast we may consider that each node provides a single resource, so to reach k resources we must reach k nodes.	[redacted]
Recently, an anycast RWA algorithm was proposed for wavelength-routed networks [33]. Anycast is a specific instance of manycast where $k=1<m$. An example of static manycast routing and wavelength can be seen in Fig. 1. The table on the right in the figure shows the static manycast request set. For each request, we must find a light-tree from the source to any two of the three destinations in the candidate set, Dc. The (optimal) RWA is shown over the six-node network on the left. Fig. 1. (Color online) Static manycast RWAexample. The requests are given in the table on the right. This RWA requires only a single wavelength. consider request 2. It is sourced at node 4, uses link 4–3, then splits the signal	[redacted]

at node 3, reaching nodes 1 and 5.

In this example, only a single wavelength is required to route all three requests over the network. We discussed the wavelength continuity constraint that specifies that a light-tree must use the same wavelength on all links. The *wavelength clash constraint* specifies that any given wavelength can be used at most once on each link.

II. PROBLEM DEFINITION

Given a network, $G = (V, E)$, a manycast request is defined as $R = (s, D_c, k)$, where $s \epsilon V$ is the source, $D_c \subseteq V- \{s\}$ is the candidate destination set, and $k \leq |D_c|$ is the number of nodes necessary to reach out of D_c. We must find a light-tree (combination of a route tree and lightpath) that starts at node s and reaches at least k nodes out of D_c. We assume that each many- cast request requires one wavelength and it can use only one wavelength .

Definition *MA-RWA(G,M)*: Given a network $G = (V, E)$ and a set of manycast requests, $M = \{R1,R2,...,Rn\}$, the solution must assign a route tree and a wavelength to each request, R_i, in such a way that the number of wavelengths required is minimized while satisfying the wavelength continuity and wavelength clash constraints.

The MA-RWA problem can be computed offline because we are given

[redacted]

[redacted]

[redacted]

the set of requests ahead of time.

A. Network Assumptions

We consider all-optical networks without wavelength converters, which implies that once the signal enters the network, it must use the same wavelength on all links (wavelength continuity constraint),(wavelength clash constraint). We assume all nodes in the network are able to split an incoming signal to any number of output ports. MC- OXCs we use in Fig. 2.

Fig. 2. MC-OXC based on splitter-and-delivery architecture.

Fig. 3. An NXN SaD switch.

our algorithms are not limited to optical cross connects. ROADMs are expected to play an integral role in all- optical networks.

B. Complexity

Routing and wavelength assignment problems to minimize the number of wavelengths are equivalent to the graph coloring problem and are therefore NP-hard [35]. Because of the complexity of the problem, we will focus on heuristic approaches to find suboptimal solutions.

III. ILP FORMULATION

We first formulate an ILP for static MA-RWA to find the optimal solution. We can still use the ILP for smaller networks to compare the results of our heuristics to the optimal solutions. The objective is to minimize the number of wavelengths used. We use *i* ,*j* to denote

[redacted]

[redacted]

[redacted]

links, m to denote the mth manycast request, and w to denote wavelengths Objective Function: *minimize: maxIndex* 1)Example: We will now provide an example of the LPH heuristic for a single manycast request. The request and network (NSFnet) are given in Fig. 4(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→2). Node 4 is added by concatenating (2→1→3→4) to the tree. This results in the tree seen in Fig. 4(b), which covers nodes 2 and 4. In the next iteration, node 4 is chosen to start the tree (0→1→3→4). A new branch can then be added to create a tree reaching node 2 (1→2), as seen in Fig. 4(c). Lastly, node 13 is chosen first with path (0→7→8→13). The tree can then be modified to branch at node 7, reaching node 4 (7→6→4), as seen in Fig. 4(d). The iterations are now complete since every node in Dc has been used as a starting node. The heurist 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	[redacted] [redacted]

is run for all requests.

Fig. 4. 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).

2) Example: We will describe an example of our move operation to generate a neighborhood set. Consider an initial set of manycast requests, $M=\{R_1,R_2,R_3,R_4\}$, where $Ri = (s_i,D_{ic}, k_j)$. The requests are first sorted in descending order according to k_i. Note, this is equivalent to sorting according to D_{ic} because, as we discuss later, we set $k_j=[D_{ic}/2]$. Let the sorted sequence now be $M' = (R_2,R_1,R_4,R_3)$. Given this sequence, the LPH heuristic is run on the requests in order to generate a solution.

During the first iteration, some percentage of the neighborhood would be explored. The entire neighborhood consists of all possible combinations generated by swapping two elements. In this simple example, we can generate six solutions by swapping: (R_1,R_2), (R_i,R_3), (R_1,R_4), (R_2,R_3), (R_2,R_4), and (R_3,R_4). With large $|M|$ values, this is too large, so instead a series of random moves are generated. Let us assume that a 50% neighborhood size was specified. This may result in the moves (R_1,R_4), (R_2,R_3), and (R_2,R_4) being randomly generated. With these moves, the neighborhood set becomes $\{(R_4,R_2,R_3,R_1),(R_1,R_3,R_2,R_4),(R_1,R_4, R_3,R_2)\}$. Given this set, LPH is run on each sequence and the best one is

[redacted]

[redacted]

chosen as the sequence to use in the next iteration (subject to the tabu list).Fig. 5. Tabu search meta-heuristic flow chart.

V. EVALUATION

We will evaluate the heuristics in two steps. First, we will compare the heuristics' results to the optimal results provided by the ILP formulation. Because of the complexity of the ILP this is only possible for small request sizes. In addition to comparing the results with the ILP, we will also compare run times.

[redacted]

Next, we will compare only the heuristics on more realistic networks with larger request sizes. We ran extensive simulations on the AT&T network, the NSF network, the Italian WDM network, and a 24-node mesh network shown in Fig. 6. We use the link distances for calculating the average tree delay, but routing is done based on hop count.

A. ILP and Heuristics Comparison

This section compares the results of the heuristics with the optimal results of the ILP. Given the complexity of the ILP, we can only run it on small networks. We use the network shown in Fig. 7. We run the ILP and heuristics for request set sizes, |M|, of 10, 15, 20, and 25. For each request set size, we ran the simulations 20 times and plot the average value (along with the 95% confidence intervals for the number of

[redacted]

wavelengths required). The source of each manycast request is uniformly distributed. The candidate destination size, (Dc), is either three or four (with equal probability) and k = 2, for all requests. The α parameter to LPH is 0.8 (best value). We used CPLEX 12.0 to obtain results for the ILP. Both the ILP and heuristics were run on a machine with a 2.33 GHz Quad Core Xeon processor and 8 GB of RAM. The processor also has Hyper-Threading, so CPLEX was able to use eight threads while solving the ILP.

The number of wavelengths required by the heuristics and ILP is shown in Fig. 8(a). The figure shows that TS provides close to optimal results and significantly outperforms SPT.

We also plot the average run times of the different solution approaches in Fig. 8(b). We can observe the large increase in run times for the ILP as the request set gets large. Note the log scale of the y-axis. The LPH and SPT heuristics essentially finish instantly compared with the others. The run time for the ILP grows rapidly. With a request set size of 25 requests over a small 6-node network, the longest run time was over 30 h for the ILP compared with under 5 min for TS.

B. Heuristics

We will first present the results obtained for our heuristics, then go on

[redacted]

to discuss how we selected the input parameters.

checked

In Table I we compare TS, LPH, and SPT. The network characteristics are given in the table where V is the number of nodes, E is number of links, δ is average nodal degree, and τ is average delay per link (ms).

Fig. 6. Networks used for heuristic evaluation

We generate a set of 150 requests. Other request set sizes provide similar patterns of results. The source node for each request is uniformly distributed over all nodes in the network. For each request *m*, the size of D_{mc} is uniformly distributed from 3 , . . . , D_{max} (a parameter representing the maximum candidate destination size) and

$k_m = [D_{mc}/2]$. Fig. 7. Six-node network used for ILP evaluation.

The destination nodes are also uniformly distributed across the network for each request. We ran each heuristic with different maximum destination set sizes, Dmax, and recorded the average number of wavelengths required, wα, and average tree delay, dα (ms). The average tree delay is defined as the average delay from the root to each destination node. The parameters for TS are as follows. There are 1000 iterations in total, the tabu tenure is 20, the fractional neighborhood search searches 6%, there are 25 iterations before diversification, and 2 diversifications before intensification (when no better

[redacted]

[redacted]

[redacted]

solution has been found for these last two cases). For both LPH and TS α=0.8 for load balancing. Most of these parameters were obtained empirically; this is discussed later in this section. Each data point is the average of 20 simulation runs. We calculated the confidence intervals but do not include them in the table. For TS and LPH, the confidence intervals were within 3% of the mean while they were slightly larger for SPT at around 5%, all with 95% confidence.	[redacted]
The table shows a significant decrease in the number of wavelengths required (wα columns) between TS and SPT as well as a significant decrease between TS and LPH. TS reduces wavelengths required by between 30% and 40% compared with SPT. The greatest gains are in the Italian network while NSFnet has the smallest gains. TS also performed about 10% better than LPH.	[redacted]
Low delay is a requirement for many next- generation applications, so the heuristics must not significantly impact delay. Even though SPT results in a smaller average tree delay (dα columns), the savings in wavelengths when using TS is significantly larger. The largest difference in delay between SPT and TS is around 1 ms. The average tree delays of TS and LPH were very similar, and TS was able to reduce wavelengths required by about 10% compared with LPH.	[redacted]
As expected, as the maximum	[redacted]

destination set size increases, the number of wavelengths required also increases. The set size of 150 was chosen for demonstration purposes. We evaluated the heuristics on varying set sizes from 50 to 200 with similar results. We chose these four networks to represent realistic scenarios with varying nodal degrees. The networks represent backbone or long-haul networks for which a wavelength-routed WDM network is a good candidate.

The relative number of wavelengths required by TS and LPH are consistent across networks (e.g., NSFnet needs the most, followed by Italy, AT&T, and the 24- node network). This is a result of the characteristics of the networks. Networks with more nodes and higher nodal degrees will require fewer wavelengths because 1) the request size is the same, so 150 requests will require fewer wavelengths on networks with more nodes than networks with fewer nodes, and 2) a network with higher nodal degree has a better chance of finding alternate paths/trees for the requests, therefore reducing the maximum number of wavelengths required on a link.

We will now discuss how we selected the parameters for our heuristics. Two decisions that affect both TS and the LPH heuristic are the choice of a for load balancing and the choice to use link distance versus hop count for shortest path routing. We found that, with the exception of the AT&T

[redacted]

[redacted]

[redacted]

network, using hop count instead of link distance had a negligible affect on average tree delay while reducing the number of wavelengths required. For the AT&T network, the delay was in some cases doubled when using hop count .Fig. 8. Performance comparison of ILP and heuristics.	[redacted]
instead of link distance. This trade-off of delay versus wavelengths required is something that must be considered for particular networks, but it seems in most cases the best choice is to perform shortest-path routing based on hop count. Shortest path based on hop count instead of link distance provides better performance because a shortest path according to link distance will likely result in paths with more hops, which results in more resource usage and therefore fewer wavelengths being available.	[redacted]
To perform load balancing, we introduce a parameter, a, where 0^a^ 1, as we discussed earlier when describing LPH. The load-balancing updates the weight of each link after a new tree and wavelength are assigned according to a+ (1-a) X c/cmax, where c is the current number of wavelengths on the link and cmax is the number of wavelengths on the most congested link. A smaller value of a puts more emphasis on load balancing when computing shortest paths. We found that if a is set too small, (e.g., 0.2), the	[redacted]

load in the network is evenly distributed over most links, but this actually increases the number of wavelengths required. One explanation is that this forces trees to get larger in size in order to use fewer loaded links, which makes it harder to find a single wavelength for later trees. Higher values of a performed the best. As we mentioned previously, we used a=0.8. This provided better distribution of load over the network than no load balancing while also decreasing the number of wavelengths required.

One disadvantage of using TS is that we must select a number of different parameters, each having different effects on the performance. To find the best combination of input parameters, we tried 81 different combinations of neighborhood fraction, tenure, diversification iterations, and intensification iterations over NSFnet with a request set size of 50. The number of iterations is fixed at 750 and a=0.8. We tried fractions of 6%, 20%, and 50%; tenure values of 10, 20, and 30; diversification values of 15, 25, and 30; and intensification values of 2, 3, and 4. We chose the parameter combination with the best cost/run time trade-off based on the empirical results.

C. Discussion

Our performance evaluation has confirmed that while manycast and multicast are similar, it is important to develop new heuristics for manycast RWA problems. Our SPT heuristic essentially treats the manycast request as a multicast request by fixing the

[redacted]

[redacted]

destination set at the source. Both LPH and TS perform much better than SPT.

While TS improves upon LPH in terms of the solution cost, TS has significantly higher run times than LPH. LPH run times are typically about a second. TS, on the other hand, can take over eight hours for a network like the AT&T network with many nodes. The TS implementation can be optimized to reduce run time, but nevertheless there is a large increase in run time for a small increase in performance. Since these computations occur offline, it may still be reasonable to allow for the longer run times to gain around a 10% improvement in cost. In any case, with realistic size networks it is not feasible to get optimal solutions unless significant computing power is available. For example, CPLEX did not find a solution after 3 days for a request set size of just 30 on a machine with 4 cores and Hyper-Threading.

VI. CONCLUSION

We have introduced the static MA-RWA problem and presented three heuristics along with an ILP to solve the problem. Our tabu search heuristic achieved between a 30% and 40% improvement over our simpler shortest-path heuristic and about a 10% improvement over LPH for realistic networks. The TS metaheuristic also produced results similar to the ILP for small networks.

[redacted]

[redacted]

We have several areas of future work for the many- cast problem over wavelength-routed networks. One is the dynamic manycast problem. This is especially applicable to Grid networks and cloud computing applications where multiple resources are required (point-to-multipoint) [14,42]. In addition to new algorithms, we could investigate extensions to our current and past work. We can use a modified LPH for each dynamically arriving request or we can also modify our distributed manycast algorithms from our work on dynamic manycast over OBS [43]. Multilayer optimization is also an interesting topic for manycast in the context of Grid networks. The selection of nodes may be based not only on the network state but also on the Grid resource utilization.	[redacted]
Another interesting area of future work is survivability of manycast requests. We can provide survivability for both link and node failures. We can protect against link failures through traditional techniques such as shared path protection, but may also be able to use the extra |Dc| - k nodes from the candidate set to either handle a link or a node failure. Switching to a different node in the candidate set depends on the type of application.	[redacted]
We are currently working on incorporating physical- layer	[redacted]

impairments into the heuristics (for static MA- RWA) in order to ensure that the signal can be received at the destinations given physical-layer impairments, such as ASE noise, crosstalk, and power loss. We have considered impairments in previous work for manycast over OBS networks [31,44]. In ad-dition to this, we have done work with QoS in dynamic manycast over OBS networks, where one of the QoS parameters is signal quality [7]. This work can be extended for wavelength-routed networks. As an extension to the work presented in this paper, we can make LPH more intelligent when generating sets of route trees by taking into account the signal quality at the nodes. We are investigating different techniques, such as providing quality of transmission (QoT) guarantee, where we use traditional RWA but do not admit connections with poor signal quality, and QoT awareness, where the RWA algorithms consider physical impairments directly.	[redacted]