Topology Aggregation for e-Science Networks

Eun-Sung Jung, Sanjay Ranka and Sartaj Sahni
Computer and Information Science and Engineering Department
University of Florida, Gainesville, FL 32611
Email: {ejung,ranka,sahni}@cise.ufl.edu

***Abstract*—We propose several algorithms for topology aggregation (TA) to effectively summarize large-scale networks. These TA techniques are shown to significantly better for path requests in e-Science that may consist of simultaneous reservation of multiple paths and/or simultaneous reservation for multiple requests. Our extensive simulation demonstrates the benefits of our algorithms both in terms of accuracy and performance.**

***Keywords*-Topology aggregation; Multi-domain routing; E-Science**

I. Introduction

The advances in communication, networking and computing technologies is dramatically changing the way scientific research is conducted. A new term, *e-Science*, has emerged to describe the "large-scale science carried out through distributed global collaborations enabled by networks, requiring access to very large-scale data collections, computing resources, and high-performance visualization" [4]. e-Science (and the related grid computing [5]) examples include high-energy nuclear physics (HEP), radio astronomy, geoscience and climate studies. To support e-Science activities, a new generation of high-speed research and education networks have been developed. These include Internet2 [6], the Department of Energy's ESnet [7], National Lambda Rail [8] etc. These networks carry a large amount of data traffic for e-Science applications.

The network supporting e-Science applications typically comprises of multiple domains. Each domain usually belongs to different organizations, and is managed based on different operational policies. In such cases, internal topologies of domains are not visible to the others for security or other reasons. However, aggregated information of internal topology and associated attributes are advertised to the other domains.

A set of techniques to aggregate data to advertise outside one domain is called *Topology Aggregation* (TA). The aggregated data itself is termed as *Aggregated Representation* (AR). A survey of TA algorithms is presented in [1]. There exists a tradeoff between the accuracy and the size of AR. Hence, most algorithms proposed in the previous work tried to achieve the most efficient AR in terms of both accuracy and space complexity.

One can classify QoS path requests into two classes: single-path single-job (SPSJ) and multiple-path multiple-job (MPMJ). SPSJ corresponds to a scenario in which all the requests consist of a single QoS path reservation. These requests are scheduled in the order of arrival. MPMJ corresponds to batch/off-line scheduling of multiple requests. These correspond to simultaneous transfer of data from multiple sources and destinations. Also, each of these requests (e.g., file transfers) can be more efficiently supported by using concurrent multiple paths.

We show that existing TA approaches developed for SPSJ do not work well with MPMJ applications as they overestimate the amount of bandwidth that is available. We propose a max flow based TA approach that is suitable for this purpose. Our simulation results demonstrate that our algorithms result in better accuracy and/or less scheduling time. TA algorithms can also be used for scheduling paths in a single domain. These methods are useful as a large domain can be partitioned into subdomains. TA algorithms can then be applied to each subdomain. With ARs on subdomains, the actual scheduling may be performed either on a single node with a rich compute resource or on a distributed set of nodes such that the time complexity of scheduling paths would be reduced by running scheduling algorithms on the partitioned smaller subdomains.

The rest of the paper is organized as follows. The related work on TA is described in Section II. Section III describes novel algorithms for MPMJ. Section IV gives time and space complexity comparison analysis. The experimental results using simulation are given in Section V, and, finally, we conclude in Section VI.

II. Related Work

TA consists of algorithms and mechanisms for reducing the size of topological information and associated attributes within a domain or subdomains while maintaining a certain level of accuracy. All TA algorithms have two elements: an aggregated graph and aggregated QoS parameter values, called *epitome*, assigned on logical links in an aggregated graph. Typical topologies used for TA are full-mesh, simple compaction, tree-based, and star-based topologies. Most TA algorithms start by building a full-mesh graph, which is a complete graph whose nodes are composed of only border nodes of the original network. Algorithms that are more focused on the size of AR usually try to transform a full-mesh graph into more compact forms, for example,

a spanning tree or a star topology, while trying to keep up with the accuracy of a full-mesh *AR*. The *epitome* is typically based on the maximum, the minimum or the average of QoS values of the subgraphs.

To the best of our knowledge, all existent TA algorithms are limited to a single QoS path routing at one time, i.e., SPSJ, with few exceptions of customized algorithms for special purposes such as computation of reliable paths. MPMJ applications consider a batch of jobs at a time and multiple paths are allowed for one job. For instance, a request for the earliest finish time for a given multiple-source multiple-destination data transfer, which is one of important e-Science applications [9], is handled at one time and multiple paths are set up for the request.

III. TA for Multiple-Path Multiple-Job (MPMJ)

A. Problem Statement

An important class of e-Science applications is bulk file transfers. For example, for high energy physics large files are routinely transferred between tiered centers that are geographically distributed around the world. The generated data have to be transferred from storage centers to research centers for the purpose of analysis or visualization. In the context of e-Science applications, bandwidth scheduling problems range from single-source single-destination data transfer optimization to multiple-source multiple-destination data transfer optimization.

The full-mesh *AR* for bandwidth scheduling, where each logical link has the maximum available bandwidth between two border nodes as an epitome, has been known as a distortion-free *AR* for single path bandwidth scheduling. However, they many have a significant degradation in accuracy for scheduling a batch of multiple jobs (each requiring multiple paths).

The computational complexity of scheduling and reserving bandwidth depends on the space requirements of the network topology. Generally, we can break down network resource provisioning procedures for e-Science applications into the admission control phase and the resource allocation phase. In admission control phase, acceptance of requested jobs is determined and then if accepted; explicit bandwidth allocation for each link will be executed in the network resource allocation phase. With compact network information abstracted from a complete network topology using topology aggregation techniques, there is a small chance that the network resource allocation phase may fail due to aggregated network status information. Although the accepted request in the admission control phase can be rejected due to inaccurate ARs in network resource allocation phase, the benefits from less space complexity and privacy of information within each domain compensates for failed operations, especially the error rate is fairly small.

In the following subsections, we propose several TA algorithms suited for MPMJ. Each request consists of single or multiple data transfer jobs. Also, we allow for the use of multiple path for bandwidth reservation. Given the large bandwidth requirements of the e-Science applications, the QoS parameter that is considered in our work is bandwidth.

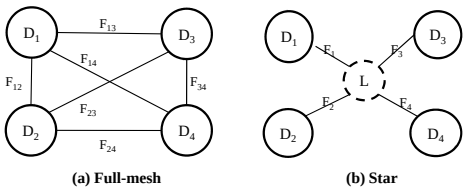

Figure 1. Full-mesh and star *AR*

B. New Topology Aggregation Algorithms

1) Full-Mesh Method: A simple way of aggregating networks with QoS parameters is by connecting every pair of nodes of interest and assigning *epitomes* to the built logical links. This results in a full-mesh topology for the nodes of interest. Consider the edge connecting nodes D_1 and D_2 in Fig. 1 (a). The epitome associated with the edge $E_{D_1D_2}$, F_{12}, may represent the max flow between the pair of nodes and can be computed using a max flow algorithm.

This simple method adapted from existent TA techniques for SPSJ may not be appropriate for MPMJ. Let us take an example of a job requesting max flow between D_1 and D_2 where D_1, D_2, D_3 and D_4 are nodes of interest. These nodes may correspond to border node in a multiple domain environment. In a single domain they may represent a small number of nodes representing a subgraph of the entire graph.

The final max flow between D_1 and D_2 may represent an overestimate for multiple simultaneous transfers (either because of concurrent transfers between multiple pairs of jobs or because of the use of multiple paths for the same transfer) as the request from $D1$ to $D3$ may also use the same edges (please recall that the edges in the AR graph do not correspond to the edges in the original graph). This leads to inaccuracy in actual scheduling.

For single path computation algorithms, several variants of the full-mesh *AR* algorithms have been proposed. They consider sparse graphs such as partial full-mesh, star, and tree for reducing the space complexity. However, if directly used, they are limited for multiple path computation algorithms.

2) Star Method: A full-mesh *AR* does not effectively support MPMJ as the maximum amount of flow that a specific node can push into a network is not restricted by other requests.

A star *AR* as in Fig. 1 (b) can overcome the drawbacks of a full-mesh *AR* by limiting the max flow value from any node. First, the logical node, L, is created and all nodes of interest are connected to it. Suppose that four nodes of interest (D_1, D_2, D_3 and D_4) are connected to the central logical node L. The epitome, assigned on the logical link connecting a certain node and the central logical node L, is a max flow value from the node to all the remaining nodes. This is easily

computed by putting a supersource node connected to a node and a supersink node connected to all the remaining nodes, and running a max flow algorithm between the supersource and the supersink nodes In this case, F_1 is the max flow value that a node D_1 can send to the network. This can be easily computed by adding a supersink node connecting D_2, D_3 and D_4 and running a max flow algorithm between D_1 and the supersink node. Likewise, we can also compute the other epitomes such as F_2, F_3 and F_4. This *AR* has only one outgoing link from each node, which keeps one node from sending the data flow beyond the epitome assigned to the outgoing links. Formal description of the algorithm is presented in Algorithm 1.

Algorithm 1 Star *AR* construction

Input: a graph $G = (V, E)$.

1: Pick nodes of interest from a full set of nodes, V.
2: Create a single logical node, L.
3: **for** each of picked nodes **do**
4: Create a link between the node and the logical node, L.
5: Compute a max flow value from a target node to all the remaining nodes.
6: Assign the computed max flow value as an epitome to the link created above.
7: **end for**

3) Partitioned Star Method: Originally, TA methods were developed to address scalability issues (in terms of space) and security issues (not exposing intradomain topology to other domains). Usually, routing procedures consist of two steps: (1) path computation and bandwidth allocation with ARs and (2) explicit path computation and bandwidth allocation with original network topology for each domain. Similar steps can also be applied for single domain network environments, where several subdomains exist for hierarchical routing or we intentionally partition one domain into several logical subdomains. In this case, the benefits from TA is almost the same as those in multi-domain network environments.

In case of MPMJ applications, an additional benefit of using the above described hierarchical approach is that we need to apply the flow algorithms for a smaller subgraph, potentially reducing the computational complexity (cf. Section IV).

The partitioned star method uses the above approach to leverage the benefits of star method by partitioning a domain into k subdomains. Each subdomain is aggregated using the star method. Fig. 2 shows an example of a domain with four partitioned subdomains.

For partitioning the graph (this has to be only done once), we use general graph partitioning algorithms [3].

Details of routing within each partition or domain are not provided due to space limitations. The reader is referred to [12] for further details.

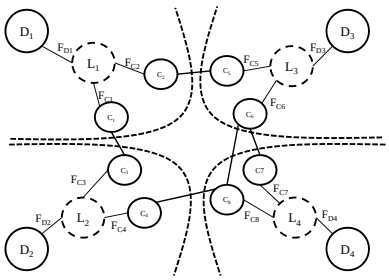

Figure 2. Partitioned star *AR*

IV. Complexity Analysis

For SPSJ algorithms, Dijkstras shortest path algorithm can be used to derive the maximum bandwidth path between two nodes. The time requirements of Dijkstra algorithm is $O(n \log n + m)$, where n is the number of vertices and m is the number of edges.

However, MPMJ algorithms require the use of max flow algorithms that have significantly higher complexity. We use the push-relabel algorithm for max-flow that has a time complexity of $O(n^3)$ [11] for our analysis. Given this, the full-mesh method and the star methods require $O(n^3D^2)$ and $O(n^3D)$ time respectively, where n is total number of nodes in the original graph and D is the number of nodes of interest.

The time requirements of the partitioned star method are considerably lower. Assuming that all the k partitions have nearly equal number of nodes, the time requirements are $O((\frac{n}{k})^3 (C + D))$, where D is the number of nodes of interest, C is the number of cut nodes, and k is the number of partitions. Thus, the partitioned star methods can potential result in significant computational benefits for graphs that are hierarchical in nature.

The space requirements of these methods and the time requirements for path computations are given in [12] due to space limitations.

V. Experiments

A. Bulk File Transfers in e-Science

We chose a bulk file transfer environment that requires the transmission of large files as the target e-Science application. In [10], we formulated the in-advance scheduling of multiple bulk file transfers as a linear programming problem. For this paper, we adapted their linear programming formulation to on-demand scheduling of multiple bulk file transfers for our simulation. Details of these methods are given in [12] due to space limitations.

B. Experiment Testbed

For TA algorithms for MPMJ, we performed experiments on random networks with a single domain. Random network topologies are generated by the BRITE internet topology generation package [2]. We tried several models such as Waxman, BRITE, etc., but the results for different models show similar trends. Therefore, we show only results for

random network topologies following the Waxman model with the average node degree of 4. The bandwidth values of edges are randomly selected from a uniform distribution between 10 to 1024. The number of nodes in each domain is varied from 100 to 300 with the increment of 50. The nodes of interest are picked randomly within a domain, and the number of nodes ranges from 1 to 16, which is doubled at each step. We generated a synthetic set of data transfer requests. Each request is described by the 3-tuple (source node, destination node, requested file transfer size). The number of requests is also randomly selected within the range of 1 to the maximum possible number of requests determined by the number of nodes of interest. For example, if the number of nodes of interest is 4, the maximum possible number of requests is 4×3. The source and destination nodes for each request are randomly selected using a uniform random number generator. The results are averaged over 100 random networks for a certain number of nodes.

C. Performance Metrics

The performance metric we have used to compare the different approaches is to find the earliest finish time (EFT) to complete all the multiple data transfer requests that are given. One would expect a good AR approach to perform as close to using the original topology. We define the error ratio (ER) as $ER = \frac{\text{TA EFT} - \text{Original EFT}}{\text{Original EFT}}$.

This ratio measures the deterioration from the applying the algorithm to determine the EFT on the original topology. Thus, a TA algorithm with lower ER is desirable.

D. Results

We measured ER according to the equation defined in Section V-C. The computational time required for each our algorithm are also recorded. Fig. 3 shows that the star and the partitioned star methods give around 5% ER. This is because the application of finding EFT tends to find and allocate all the available bandwidths in a network, which are limited by the star or the partitioned star ARs in a similar way as the original network does. In addition, we observe that as the number of requests increase, ER improves. This is because all the network resources, i.e., the bandwidths, are eventually used up and there is no difference between applying AR and not applying AR. The performance of full-mesh *AR* was considerably lower than the other two algorithms. Also, the star method is comparable to the partitioned star method in terms of accuracy (Fig. 4).

Surprisingly, the partitioned approach did not provide any computational benefits (in fact the time requirements were significantly higher). We believe that this is mainly due to the fact that the networks had relatively random topologies and the number of cut nodes was high. We expect that if the domain is hierarchical, the number of cut nodes will be lower, and potentially will enhance the performance of the partitioned star method.

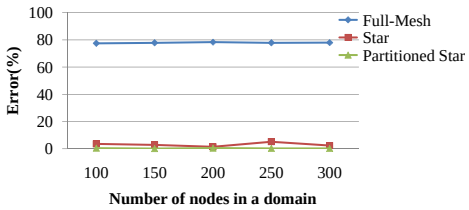

Figure 3. Error ratio vs. the number of nodes

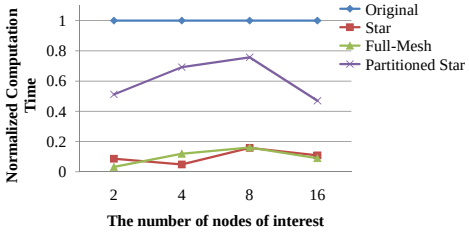

Figure 4. Normalized computational time vs. the number of source and destination nodes

VI. Conclusions

We propose several algorithms for topology aggregation (TA) to effectively summarize large-scale networks. These TA techniques are shown to significantly better for path requests in e-Science that may consist of simultaneous reservation of multiple paths and/or simultaneous reservation for multiple requests. Our extensive simulation demonstrates the benefits of our algorithms both in terms of accuracy and performance. The proposed algorithms, star and partitioned star, are shown to be significantly better than existing approaches in terms accuracy. Thus, it is well suited for e-Science applications that require reservations of multiple paths in multiple domains.

Acknowledgment

This work was supported, in part, by the National Science Foundation under grant 0312038 and 0622423. Any findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

References

[1] Suleyman Uludag, King-Shan Lui, Klara Nahrstedt and Gregory Brewster, Analysis of Topology Aggregation techniques for QoS routing, ACM Computing Surveys, 39, 2007.
[2] A. Medina, A. Lakhina, I. Matta and J. Byers, BRITE: an approach to universal topology generation, Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, 346–353, 2001.
[3] George Karypis and Vipin Kumar, MeTis: Unstrctured Graph Partitioning and Sparse Matrix Ordering System, Version 2.0, www.cs.umn.edu/ metis, 1995.
[4] The U.K. Research Councils, http://www.research-councils.ac.uk/escience/, site last visited on Feb. 18, 2008".
[5] I. Foster and C. Kesselman, The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann, 1999.
[6] Internet2, http://www.internet2.edu.
[7] Energy Science Network (ESnet), http://www.es.net.
[8] National Lambda Rail, http://www.nlr.net.
[9] Tiziana Ferrari, Grid Network Services Use Cases from the e-Science Community, Dec. 2007.
[10] Yan Li and Sanjay Ranka and Sartaj Sahni, In-advance path reservation for file transfers In e-Science applications, Computers and Communications, IEEE Symposium on, 176–181,July, 2009.
[11] Ravindra Ahuja, Network flows : theory, algorithms, and applications, Prentice Hall, 1993.
[12] Eun-Sung Jung, Sanjay Ranka, and Sartaj Sahni, Topology Aggregation for e-Science Networks, *CISE Department, University of Florida, Technical Report*, 2010,