Topology Aggregation for e-Science Networks

EUN-SUNG JUNG, SANJAY RANKA AND SARTAJ SAHNI
Computer and Information Science and Engineering Department
University of Florida
{ejung,ranka,sahni}@cise.ufl.edu

We propose several algorithms for topology aggregation (TA) to summarize large-scale networks that are becoming prevalent in e-Science. These TA techniques are shown to be 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: Design, Experimentation, Measurement, Performance, Reliability

1. INTRODUCTION

The advance of communication, networking and computing technologies is dramatically changing the ways how 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" [UKescience]. Well-quoted e-Science (and the related grid computing [Foster et al. 1999]) 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 [Internet2], the Department of Energy's ESnet [ESnet], National Lambda Rail [Lrail] etc. These networks carry a large amount of data traffic for e-Science applications.

The need for transporting large volumes of data in e-Science has been well-argued [Newman et al. 2003; Bunn et al. 2003]. For instance, the HEP data is expected to grow from the current petabytes (PB) (10^{15}) to exabytes (10^{18}) sometime between 2012 to 2015. In addition, e-Scientists desire schedulable network services to support predicable work processes [Ferrari 2007]. Quality of service (QoS) in network applications has been an active research area for several decades. Recently new technologies such as multiprotocol label switching (MPLS) and generalized multiprotocol label switching (GMPLS) drew more attention to QoS routing since those technologies have made it possible for network managers to set up and tear down explicit paths while guaranteeing specified amounts of bandwidth.

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 may not be visible to the others for security or other reasons. Instead, aggregated information of internal topology and associated attributes is 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 [Uludag et al. 2007]. 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 or less scheduling time.

BGP, which has been deployed for inter-domain protocol, has limited use for AR techniques, as it is not flexible enough to be extended to accommodate many QoS parameters. This is because it was originally designed only for distributing reachability information [Yannuzzi et al. 2005]. Recently a new network model based on path computation elements (PCEs) has been proposed to overcome the aforementioned drawbacks of BGP [Farrel]. PCE is an entity that is capable of computing network paths utilizing the traffic engineering database which contains required network status information such as a topology, available bandwidth on links and etc. Recent papers [Ricciato et al. 2005; Pelsser et al. 2006; Sprintson et al. 2007] have based their network model on PCE-based architecture. We develop TA algorithms in the context of PCE-based architecture that can support most e-Science applications. In particular, the following network model is assumed throughout the paper.

(1) A centralized PCE exists per each domain. A node sends a request to the PCE to make a reservation for a QoS path.
(2) Centralized PCEs flood aggregated topology information to others so that every centralized PCE maintains a complete view of a network represented by ARs except its own domain.

The first condition states that one active element in a domain acts as a supernode in the domain, which knows every information essential for QoS path computation. One possible implementation is that every node in a domain send a request for QoS path to the designated centralized PCE, therefore, the PCE can manage one consistent information on network status related to QoS parameters. The second condition can be reasonably assumed in e-Science networks, of which size is relatively very small compared to the Internet. This statement enables us to directly apply QoS routing algorithms which have been developed so far. In this network architecture, one domain can advertise its aggregated topology information and associated QoS parameters to all the other domains.

Based on the described network model, a scenario of inter-domain QoS routing works as in Figure 1.

- ➢ **STEP 1** A source node sends a path computation request to a single centralized PCE in the same domain.
- ➢ **STEP 2** Then the PCE replies back with a *coarse path*, which consists of a sequence of border nodes without detailed hops between border nodes.
- ➢ **STEP 3** With the *coarse path*, the source node sends a path setup request that will traverse border nodes of the *coarse path*.
- ➢ **STEP 4 and 5** The border node which receives a path setup request gets a *strict path* for a *coarse path* from the PCE in the same domain. The *strict path* contains the detailed hop information within the domain.
- ➢ **STEP 6** The same steps repeat until a path setup request reaches a destination node.

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 2. Section 3 describes novel algorithms for MPMJ. Section 4 describes how real routing works

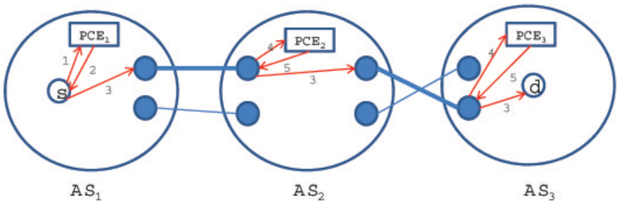


Figure. 1: An example of inter-domain QoS routing

with TA algorithms, and Section 5 gives time and space complexity comparison analysis. The experimental results by simulation are given in Section 6, and, finally, we conclude in Section 7.

2. 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. Uludag et. al [Uludag et al. 2007] presented a survey of these algorithms for multi-domain environments. All TA algorithms have two elements: an aggregated graph and aggregated QoS parameter values, called *epitome*, assigned on logical links in an aggregated graph.

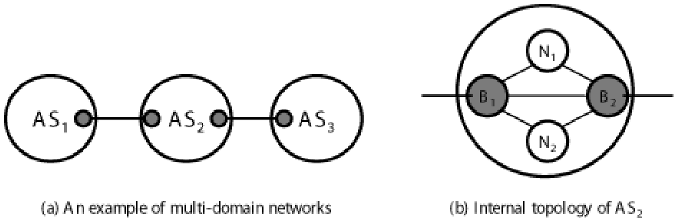
Typical topologies used for TA are full-mesh, simple compaction, and star-/tree-based topologies. Some other topologies, e.g., Shufflenet [Yoo et al. 2000], have been proposed to reduce space complexity in specific cases such as asymmetric networks. Most TA algorithms start by building a full-mesh graph, 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.

TA algorithms for SPSJ in large-scale multi-domain networks focus on the compaction of ARs – accuracy is not the top priority. As for TA algorithms in small sized networks, accuracy has been the main focus [Sarangan et al. 2004; Ricciato et al. 2005; Pelsser et al. 2006; Sprintson et al. 2007]. For a single QoS constraint, a distortion-free algorithm exists [Uludag et al. 2007]. But for two QoS constraints composed of an additive and a restrictive one, the problem gets more complicated. Even though the problem itself is not intractable, distortion-free representation is not compact. For such reasons, several approximating algorithms minimizing distortion , e.g., the line segment algorithm [Lui et al. 2004], have been proposed. Usually, the multiple QoS constraints problem is generalized as one restrictive with multiple additive constraints, since a multiplicative constraint such as a link reliability can be transformed into an additive one through a *log* operation.

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 [Ferrari 2007], is handled at one time and multiple paths are set up for the request.

The emerging technologies such as MPLS or GMPLS make it possible that applications requiring strict QoS requirements are implemented on networks equipped with such facilities. Special purpose networks such as research networks linking national labs in the US can be set up to support those applications [Rao et al. 2005]. Especially for inter-domain QoS path routing in such special purpose networks, the accuracy of aggregated topologies and associated QoS parameter values is more important than the size of data exchanged among domains since the number of domains is relatively small compared to the Internet which is constituted by a huge number of hosts and switches. Thus the need for more accurate ARs is prominent.

One of the most recent work regarding TA for two QoS constraints is the line segment algorithm in delay-bandwidth sensitive networks [Lui et al. 2004]. The line segment algorithm first computes 2-D charts whose x-axis and y-axis are delay and bandwidth respectively, for every pair of border nodes. The chart contains all the information for computing QoS paths with delay and bandwidth constraints. Authors in [Lui et al. 2004] suggested the line segment algorithm approximating those information by a line to reduce the size of data representing all possible delay-bandwidth combinations between two border nodes, and it is possible because the shape of the charts takes a increasing staircase function. The next step is to establish a full-mesh topology and convert it to a star topology to further enhance the space complexity up to $O(|B|)$.



(a) An example of multi-domain networks

(b) Internal topology of AS_2

Figure. 2: An illustrative example for limitations of the line segment algorithm

With existent TA algorithms for SPSJ, there is no way to estimate if more than one path between two border nodes are available. Consider a multi-domain network in Figure. 2. The network consists of three domains/autonomous systems (ASes) where AS_1 is connected to AS_3 via AS_2. Suppose that a host in AS_1 wants to find max flow paths or reliable paths, composed of a primary and a backup path, to a certain host in AS_3. If TA algorithms such as the line segment algorithm is deployed in this network, the PCE in AS_1 computes paths based on the AR from AS_2, which only gives the information on how much bandwidth is available within a certain delay. Since the PCE in AS_1 has no clue on how many paths exists internally in AS_2, the computed max-flow or reliable paths are likely to be biased compared to the most accurate paths computed based on the complete network status information.

3. TA FOR MULTIPLE-PATH MULTIPLE-JOB (MPMJ)

3.1 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 may 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 compensate for failed operations, especially when 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.

3.2 New Topology Aggregation Algorithms

3.2.1 *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 Figure. 3. The epitome associated with the edge $E_{D_1 D_2}$, F_{12}, may represent the max flow between the pair of nodes and can be computed using a max flow algorithm. The algorithm for building a full-mesh AR is described in Algorithm 1.

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 nodes 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.

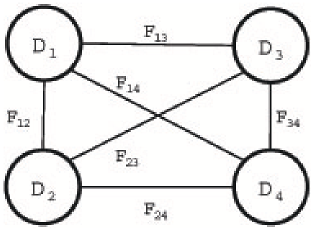


Figure. 3: Full-mesh AR

Algorithm 1 Full-mesh AR construction

```
Input: a graph $G = (V, E)$.
Pick nodes of interest from a full set of nodes, $V$, and add them to the AR.
for each pair of picked nodes do
    Create a link between two nodes
    Compute a max flow value between two nodes.
    Assign the computed max flow value as an epitome to the link created above.
end for
```

3.2.2 *Star Method.* A full-mesh AR does not effectively support MPMJ as the maximum amount of flow that a certain node can push into a network is not restricted. For single path computation algorithms, most recent TA techniques start from full-mesh AR and produce diverse variants stemming from it such as partial full-mesh, star, tree and so on. For multiple path computation algorithms, the reasons described in the previous subsections prevents full-mesh AR from being utilized as a base AR for other efficient ARs in terms of space complexity.

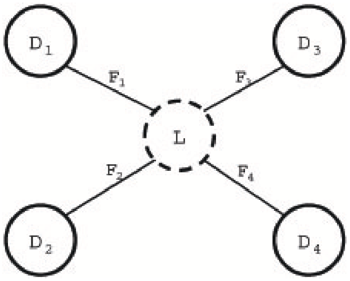


Figure. 4: Star AR

A star AR as in Figure. 4 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 a max flow value that a node D_1 can send to the network, which is 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 link. Formal description of the algorithm is presented in Algorithm 2.

Algorithm 2 Star AR construction

```
Input: a graph $G = (V, E)$.
Pick nodes of interest from a full set of nodes, $V$, and add them to the AR.
Create a single logical node, $L$.
for each picked nodes do
    Create a link between the node and the logical node, $L$.
    Compute a max flow value from a target node to all the remaining nodes.
    Assign the computed max flow value as an epitome to the link created above.
end for
```

3.2.3 *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 are 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 5).

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. Figure. 5 shows an example of a domain with four partitioned subdomains. We call the nodes and edges connecting partitioned subdomains, e.g., c_1 and c_3, and E_{c_1,c_3}, *cut nodes* and *cut edges*, respectively.

In this paper, we use general graph partitioning algorithms, which are widely used in many other computer science areas including load distribution in parallel computers, sparse matrices and design of very large scale integrated circuits (VLSI) [Karypis et al. 1995]. The algorithm for building partitioned star AR is described in Algorithm 3.

Algorithm 3 Partitioned star AR construction

Input: A graph $G = (V, E)$ and k, the number of partitions (subdomains).

```
Pick nodes of interest from a full set of nodes, $V$ and add them to the AR.
Partition a graph into $k$ parts so that the number of nodes of interest is evenly distributed over partitioned parts.
Identify cut nodes and cut edges, and add them to the AR.
for each subdomain do
    Construct star $AR$ with picked nodes and cut nodes in the subdomain.
end for
```

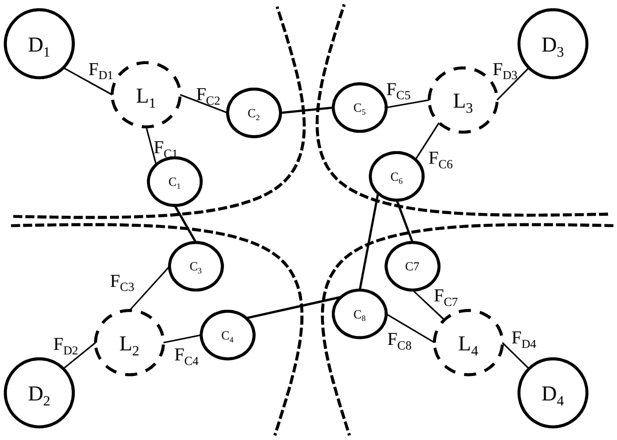


Figure. 5: Partitioned star AR

4. ROUTING

With the network model described in Section 1, inter-domain QoS path routing is relatively easy compared to a QoS path routing using distance vector routing protocols. Any centralized PCE can compute a path to a destination which consists of a strict path within its own domain and a coarse inter-domain path to the destination domain. The coarse inter-domain path is composed of border nodes, and when the path setup request is received by a border node on the intermediate path, it is translated into a strict path composed of intra-domain routers or switches.

Inter-domain routing for SPSJ applications is well described in Section 1. The routing procedures for MPMJ applications are the same as those for SPSJ applications. The results from any algorithms, e.g., a max flow path algorithm, run on ARs are expanded on each domain or each subdomain by running the same algorithm on the original topology of a domain or a subdomain. If operations fail in any of the domains or subdomains, the entire operation will fail. Note that the reason MPMJ applications in intradomain environments use ARs of subdomains is to reduce the time complexity of scheduling, whereas SPSJ or MPMJ in interdomain environments are forced to use ARs for security or administrative reasons. The benefits of using ARs in intradomain environments from the perspective of time complexity will be described in Section 5.

5. COMPLEXITY ANALYSIS

For SPSJ algorithms, Dijkstraś 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 $O(n^3)$ [Ahuja 1993] for our analysis. Given this, the full-mesh method and the star methods require $O(n^3 D^2)$ and $O(n^3 D)$ 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 $O(\left(\frac{n}{k}\right)^3 (C + D))$, 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 potentially result in significant computational benefits for graphs that are hierarchical in nature. The time complexities of TA algorithms for MPMJ are summarized in Figure. 6.

Method	Time Complexity
Full-mesh	$O(n^3 D^2)$
Star	$O(n^3 D)$
Partitioned star	$O(\left(\frac{n}{k}\right)^3 (C + D))$

D = number of nodes of interest
C = number of cut nodes
k = number of partitions

Figure. 6: Time Complexity for MPMJ

The space complexities are summarized in Figure. 7. The space complexities of ARs for full-mesh, star and partitioned star methods are $O(D^2)$, $O(D)$ and $O(C + D)$, respectively. Suppose that a certain algorithm for MPMJ applications takes $O(n^3)$. If we run the algorithm on partitioned star ARs, it will take $O((C + D)^3)$ and $kO(\left(\frac{n}{k}\right)^3)$, which are time taken for running the algorithm on ARs and time taken for explicit routing in each partition, respectively. $(C + D)$ and k is definitely a small value compared to n, and n^3 may be greater than $\left(\frac{n}{k}\right)^3$ in a few orders of magnitude. Hence, we can expect that the partitioned star method can expedite the path computation and bandwidth allocation process significantly.

6. EXPERIMENTS

6.1 Bulk File Transfers in e-Science

We chose a bulk file transfer application in [Ranka et al. 2009] as a typical MPMJ e-Science application to show that our proposed algorithms perform better than naive algorithms adapted from SPSJ TA algorithms. In [Ranka et al. 2009], authors formulated the in-advance scheduling

Method	Space Complexity
Full-mesh	$O(D^2)$
Star	$O(D)$
Partitioned star	$O(C + D)$

D = number of nodes of interest
C = number of cut nodes

Figure. 7: Space Complexity for MPMJ

of multiple bulk file transfers as a linear programming problem. We adapted their linear programming formulation to on-demand scheduling of multiple bulk file transfers for our simulation. The linear programming formulation is shown in Figure 8. The notations and equations are borrowed from [Ranka et al. 2009] whenever possible. In this formulation, t_f denotes the time by which all file transfers complete. The objective of this linear programming problem is to find the earliest finish time. f_{lk}^{j} is the amount of file transferred for request $j \in F$ on link $(l, k) \in E$. b_{lk} is the bandwidth available on link (l, k). Equation 3 ensures that for each transfer request $j \in F$, for each node l that is neither the source nor the destination node, the amount of file j that leaves node l equals the amount that enters this node. Equation 4 requires the source node of request j to send a net f_j units of file j out and requires the destination node to receive a net f_i units. Equation 5 ensures that the amount of traffic on each link does not exceed the available capacity of any link in the interval $[0, t_f)$. Equation 6 ensures that file transfer amounts are non-negative.

$$\text{minimize } t_f \tag{1}$$

$$\text{subject to} \tag{2}$$

$$\sum_{k:(l,k)\in E} f_{lk}^{j} - \sum_{k:(k,l)\in E} f_{kl}^{j} = 0$$

$$\forall j \in F, \forall l \in V, l \neq s_j, l \neq d_j \tag{3}$$

$$\sum_{k:(l,k)\in E} f_{lk}^{j} - \sum_{k:(k,l)\in E} f_{kl}^{j} =$$

$$\begin{cases} f_j, & \text{if } l = s_j \\ -f_j, & \text{if } l = d_j \end{cases}, \forall j \in F \tag{4}$$

$$\sum_{j\in F} f_{lk}^{j} \leqslant b_{lk} \times t_f, \forall (l,k) \in E \tag{5}$$

$$f_{lk}^{j} \geqslant 0 \tag{6}$$

Figure. 8: Earliest finish time on-line scheduling of multiple file transfers

6.2 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 [Medina et al. 2001]. 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 2 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.

6.3 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.

Hence, we use the the error ratio (ER) that measures the deterioration from the correct EFT on the original topology. A TA algorithm with lower ER shows better performance. ER is formally defined as

$$ER = \frac{\text{TA EFT} - \text{Original EFT}}{\text{Original EFT}}$$

6.4 Simulation Results and Discussion

We measured ER according to the equation defined in Section 6.3. The computational times taken for each our algorithm are also recorded to show how much computation cost reduction we can get from the compact representation. Figure. 9 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 is improved because all the network resources, i.e., the bandwidths, are eventually used up. As related work, authors in [Grimmet et al. 1982] showed that the minimum cut for the complete graph with independent and identically distributed (i.i.d.) edge capacities is almost surely the set of edges incident on the source or the set of edges incident on the destination. In our experiments, the connectivity of nodes are set to average degree of 4, and there exists a tendency that ER is improved as the number of requests increase. This shows that the maximum topology capacity with regard to multiple max flows between nodes is well captured by the star AR as the number of transfer increases. In future, we will analytically study the results based on previous work on max flows in random networks.

As expected, the performance of full-mesh AR is the worst. 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

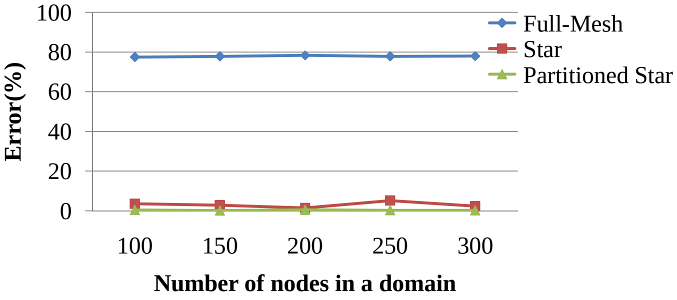


Figure. 9: Error ratio vs. the number of nodes

Surprisingly, the partitioned approach did not provide any computational benefits as shown in Figure. 10 (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.

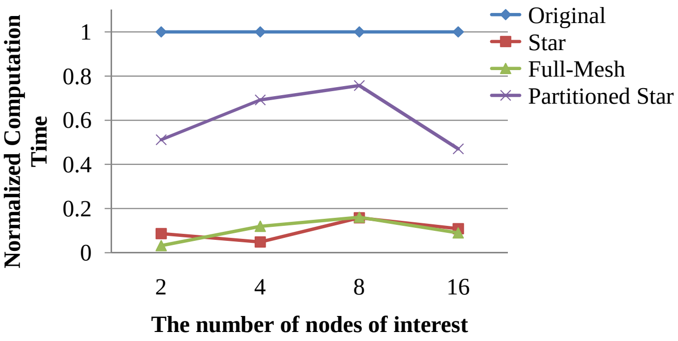


Figure. 10: Normalized computational time vs. the number of source and destination nodes

7. 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 of accuracy. Thus, it is well suited for e-Science applications that require reservations of multiple paths in multiple domains.

REFERENCES

Abilene *http://abilene.internet2.edu/*.

Baruch, A., and Yuval, S. 2001. Topology aggregation for directed graphs, *IEEE/ACM Transactions on Networking*, 9, 82–90.

Bunn, J., and Newman, H. 2003. Data-intensive grids for high-energy physics, *Grid Computing: Making the Global Infrastructure a Reality*, John Wiley & Sons, Inc.

CA*net4 *http://www.canarie.ca/canet4/index.html*

Energy Science Network (ESnet) *http://www.es.net*

Ferrari, T. Dec 2007. Grid Network Services Use Cases from the e-Science Community.

Farrel, A. A Path Computation Element (PCE)-Based Architecture *http://www.ietf.org/rfc/rfc4655.txt*

Foster, I., and Kesselman, C. 1999. The Grid: Blueprint for a New Computing Infrastructure, Morgan Kaufmann.

Geant2 *http://www.geant2.net*

Grimmett, G.R., and Welsh, D.J. 1982. Flow in networks with random capacities, *Stochastics*, 205-229.

Internet2 *http://www.internet2.edu*

Johnston, W.E.,Metzger, J. ,OConnor, M. ,Collins, M. ,Burrescia, E.,Dart, E.,Gagliardi, J.,Guok, C., and Oberman, K. 2008. Network Communication as a Service-Oriented Capability, *High Performance Computing and Grids in Action*, Vol. 16.

Karypis, G., and Vipin, K. 1995. MeTis: Unstrctured Graph Partitioning and Sparse Matrix Ordering System, Version 2.0, *www.cs.umn.edu/ metis.*

King-Shan Lui, Nahrstedt, K., and Chen, S. 2004. Routing with topology aggregation in delay-bandwidth sensitive networks, *IEEE/ACM Transactions on Networking*, 12, 17–29.

Korkmaz, T., and Krunz, M. 2000. Source-oriented topology aggregation with multiple QoS parameters in hierarchical networks, *ACM Transactions on Modeling and Computer Simulation*, 10, 295–325.

National Lambda Rail,*http://www.nlr.net*

Liu, J.,Niu Z., and Zheng J. 2000. Parameter Dimensioning Algorithms of the PNNI Complex Node Model with Bypasses, *IEICE Transaction on Communication*, E83-B, 638–645.

Medina, A., Lakhina, A.,Matta, I., and Byers, J. 2001. BRITE: an approach to universal topology generation,Ninth International Symposium on Modeling, *Analysis and Simulation of Computer and Telecommunication Systems*, 346–353.

Newman, H. B.,Ellisman, M. H., and Orcutt, J.A. Nov 2003. Data-intensive e-Science frontier research, *Communications of the ACM*, 46, 11,68–77.

Pelsser, and Bonaventure 2006. Path Selection Techniques to Establish Constrained Interdomain MPLS LSPs, *Networking Technologies, Services, and Protocols*, 209–220.

Rao, N.S.,Carter, S.M.,Wu, Q.,Wing, W.R.,Zhu, M.,Mezzacappa, A.,Veeraraghavan, M. , and Blondin, J.M. 2005. Networking for large-scale science: Infrastructure, provisioning, transport and application mapping,*Proceedings of SciDAC Meeting.*.

Ravindra, A. 1993. Network flows : theory, algorithms, and applications, *Prentice Hall.*

Ricciato,F.,Monaco, U., and Ali, D. 2005. Distributed schemes for diverse path computation in multidomain MPLS networks, *IEEE Communications Magazine*, 43, 138–146

SOHEILI, A.,KALOGERAKI, V., and GUNOPULOS, D. 2005. Spatial queries in sensor networks. In *Proceedings of the Proceedings of the 13th annual ACM international workshop on Geographic information systems, Bremen, Germany 2005 ACM.*

Sarangan, V.,Ghosh, D., and Acharya, R. 2004. Performance analysis of capacity-aware state aggregation for inter-domain QoS routing, *IEEE Global Telecommunications Conference*, 3, 1458–1463.

Sprintson, A.,Yannuzzi, M.,Orda, A., and Masip-Bruin, X. 2007. Reliable Routing with QoS Guarantees for Multi-Domain IP/MPLS Networks,*IEEE International Conference on Computer Communications*, 1820–1828.

Tam, W.-Y.,Lui, K.-S., Uludag, S. , and Nahrstedt, K. Aug 2007. Quality-of-Service routing with path information aggregation, *Computer Networks*, 51, 3574–3594.

UKescience The U.K. Research Councils, *http://www.research-councils.ac.uk/escience/* (Accessed: Feb 2008)

Uludag, S.,Lui, K.-S. ,Nahrstedt, K., and Brewster, G. 2007. Analysis of Topology Aggregation techniques for QoS routing,*ACM Computing Surveys*, 39.

Li, Yan,Ranka, S., and Sahni, S. July 2009. In-advance path reservation for file transfers In e-Science applications, *Computers and Communications, 2009. ISCC 2009. IEEE Symposium on*, 176–181.

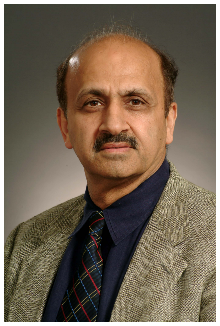
Yannuzzi, M.,Masip-Bruin, X., and Bonaventure, O. 2005. Open issues in interdomain routing: a survey, *IEEE Network*, 19, 49–56.

Tang, Y. , and Chen, S. 2004. QoS information approximation for aggregated networks, *IEEE International Conference on Communications*, 4, 2107–2111.

Yoo, Y.,Ahn, S. , and Chong Sang Kim 2000. Link state aggregation using a shufflenet in ATM PNNI networks, *IEEE Global Telecommunications Conference*, 481–486.

Eun-Sung Jung is a Ph. D. Student in the Department of Computer and Information Science and Engineering at the University of Florida. He received B.S. and M.S. degrees in electrical engineering from Seoul National University, Korea, in 1996 and 1998, respectively. His research interests include network optimization in connection-oriented networks and its applications to existing research networks. He published several papers in conference proceedings.

Sanjay Ranka is a Professor in the Department of Computer Information Science and Engineering at University of Florida. His current research interests are energy efficient computing, high performance computing, data mining and informatics. Most recently he was the Chief Technology Officer at Paramark where he developed real-time optimization software for optimizing marketing campaigns. Sanjay has also held positions as a tenured faculty positions at Syracuse University and as a researcher/visitor at IBM T.J. Watson Research Labs and Hitachi America Limited. Sanjay earned his Ph.D. (Computer Science) from the University of Minnesota in 1988 and a B. Tech. in Computer Science from IIT, Kanpur, India in 1985. He has coauthored two books: Elements of Neural Networks (MIT Press) and Hypercube Algorithms (Springer Verlag), 70 journal articles and 110 refereed conference articles. He is a fellow of the IEEE and AAAS, and a member of IFIP Committee on System Modeling and Optimization. He serves on the editorial board of the Journal of Parallel and Distributed Computing.



Sartaj Sahni is a Distinguished Professor and Chair of Computer and Information Sciences and Engineering at the University of Florida. He is also a member of the European Academy of Sciences, a Fellow of IEEE, ACM, AAAS, and Minnesota Supercomputer Institute, and a Distinguished Alumnus of the Indian Institute of Technology, Kanpur. In 1997, he was awarded the IEEE Computer Society Taylor L. Booth Education Award "for contributions to Computer Science and Engineering education in the areas of data structures, algorithms, and parallel algorithms", and in 2003, he was awarded the IEEE Computer Society W. Wallace McDowell Award "for contributions to the theory of NP-hard and NP-complete problems". Dr. Sahni was awarded the 2003 ACM Karl Karlstrom Outstanding Educator Award for "outstanding contributions to computing education". Dr. Sahni received his B.Tech. (Electrical Engineering) degree from the Indian Institute of Technology, Kanpur, and the M.S. and Ph.D. degrees in Computer Science from Cornell University. Dr. Sahni has published over three hundred research papers and written 15 texts. His research publications are on the design and analysis of efficient algorithms, parallel computing, interconnection networks, design automation, and medical algorithms. Dr. Sahni is a co-editor-in-chief of the Journal of Parallel and Distributed Computing, a managing editor of the International Journal of Foundations of Computer Science, and a member of the editorial boards of Computer Systems: Science and Engineering, International Journal of High Performance Computing and Networking, International Journal of Distributed Sensor Networks and Parallel Processing Letters.