Robust Application Specific and Agile Private (ASAP) Networks Withstanding Multi-layer Failures

Xin Liu[*], Chunming Qiao[*], Ting Wang[†]

[*]*Department of Computer Science and Engineering, State University of New York at Buffalo, New York 14260*

[†]*NEC Laboratories America, Princeton, NJ 08540*

xliu8@cse.buffalo.edu

Abstract: A novel robust network design framework supporting critical distributed computing applications over WDM networks is studied, taking into consideration concurrent and multi-layer failures and their impact on a large number of distributed computing applications.

OCIS codes: (060.4510) Optical communication; (060.4250) Networks

1. Introduction

Modern distributed computing applications under the new service oriented architecture (SOA) or Cloud Computing require both executions by multiple geographically separated computing clusters (each of which could provide CPU processing, data storage or visualization display capabilities) and high-speed data transferring over fibers between them. Each instance of these distributed applications (or a job) may be logically described by a task graph, where a vertex represents a task and an edge represents the communication between two tasks. The primary challenge of supporting these applications is to find an optimal mapping from the task graph (an overlay) to the substrate network such as a WDM network connecting many computing clusters. For each accepted job request, the set of clusters chosen to execute the tasks, and the lightpaths established among them during the task execution, together form what we call an *Application Specific and Agile Private* (ASAP) network. Since different clusters may be chosen to execute the tasks and each lightpath has a limited duration, such an ASAP network differs from a conventional "virtual network" which usually refers to a collection of lightpaths with unlimited duration connecting a fixed set of end points (e.g., IP routers in an IP/WDM networks). In addition, one must deals with a large number of ASAP networks, each with a limited lifetime, instead of only one or a few virtual networks of a relatively long duration.

For most of the medical, financial, military and homeland security applications, the ability to survive fiber cuts or computer cluster failures is critical. Several studies have been conducted assuming a single failure or multiple failures at the WDM layer, and there are also a few studies on coordination among multiple layers (e.g., IP and WDM) after a failure occurs [1-3]. However, none has studied the case where multiple failures can affect multiple layers when supporting distributed applications and in particular, none has studied the case with concurrent failures of a computing cluster (due to e.g., power outage or virus attack), and lightpaths (due to e.g., a fiber cut). For example, the objective in [3] is to route the lightpaths that constitute the virtual topology over the WDM network so that the surviving nodes of the original virtual topology are still connected after fiber failures, but with a different virtual topology (with a reduction in the connectivity and bandwidth). In contrast, our objective is, after a computing cluster failure and a lightpath failure, to find a replacement computing cluster and additional lightpaths so as to provide the same ASAP as before to the affected distributed computing application.

In the following, the novel framework for survivable ASAP networks is devised to provide failure recovery for distributed applications from multilayer failures. In particular, two novel resource allocation schemes within the framework are proposed and compared via simulation using the job blocking rate (the ratio of the number of blocked jobs to the number of all simulated jobs) and the average resource leasing cost as the performance metric.

2. Models and Assumptions

2.1 Network Model

The WDM network with attached computing clusters can be formulated as an edge-weighted undirected graph $G_n = (\mathbf{V}, \mathbf{E}, \mathbf{r}, \mathbf{h}, \mathbf{w})$. In this study, we assume every node v in $\mathbf{V}$ consists of a WDM switch and an edge node, which in turn is connected to one computing cluster. Each cluster connecting to such a node v provides a single type of computing resources denoted by $\mathbf{r}_v$, which could be either processing, storage or display. Let $\mathbf{h}_v$ and $\mathbf{w}_e$ be the capacity of the computing cluster (in terms of the number of CPUs) attached to node v and the number of available wavelengths on link e in $\mathbf{E}$ respectively.

2.2 Task Graph

A task graph is commonly used to describe the communication requirement among tasks of a job. A task graph can be formulated as $G_t = (\mathbf{N}, \mathbf{C}, \mathbf{r}, \mathbf{q}, \mathbf{b})$ [4]. Each node n in $\mathbf{N}$ represents a task and a connection c in $\mathbf{C}$ indicates that the corresponding two tasks communicate with each other one or more times during the task execution. Accordingly, an ASAP network is formed by establishing $2|\mathbf{C}|$ lightpaths for each job between assigned computing clusters. Note

978-1-55752-865-0/09/$25.00 ©2009 IEEE

that here, the clusters are connected over lightpaths due to the need for high bandwidth and service isolation/guarantee, but our concept applies to subwavelength connections as well. Let $\mathbf{r}_n$ denote the type of the resources required by task n and a task can only be executed on the computing clusters providing the same resources required by the task. For each task n, let $\mathbf{q}_n$ denote its required number of CPUs. For each connection c, let $\mathbf{b}_c$ denote the estimated average number of wavelengths required between two tasks connected by c.

2.3 Cost Model

In this work, we consider the objective being the minimization of the resource usage/leasing cost. More specifically, there is an execution cost $\mathbf{e}_{nv}$ of executing task n on cluster v, which is a function of the computing requirement $\mathbf{q}_n$ of task n. The communication cost between cluster v (executing task n) and cluster v' (executing task n') is assumed to be $g\mathbf{b}_{nn'}\mathbf{p}_{vv'}$, where $\mathbf{b}_{nn'}$ is the bandwidth requirement between two tasks n, n' and $\mathbf{p}_{vv'}$ is the length of the path connecting v and v'. We use a parameter g to govern the ratio of unit communication cost to unit execution cost.

3. Robust ASAP Network Design Framework

3.1 A Two-step Mapping Approach

We take a two-step approach to the complicated survivable assignment problem. In the first step, the task graph is mapped to an ASAP topology with certain resource isolation constraints. In the second step, the ASAP topology is mapped to the substrate network. To illustrate the two-step approach, we assume a simple mission-critical job whose task X is assigned to cluster A, and needs to communicate with another task Y assigned to cluster B, as shown in Figure 1(a). Suppose we consider two concurrent failures, one at a cluster (A or B), and the other a link failure that could occur anywhere within the network. We can have several ways to provision robust virtual networks. In Figure 1(b) (where the links are virtual links, which are paths in the substrate network), it is sufficient if disjoint paths exist between A and B, A' and B' respectively. Note that other pairs of paths can share common links to increase resource utilization as shown in (c) (where a thick line segment indicates the overlapping of physical links). Alternatively, we can use an ASAP topology in (d) where three disjoint virtual networks are provisioned. Here, by "disjoint virtual networks", we mean multiple virtual networks do not share any common links and computing clusters.

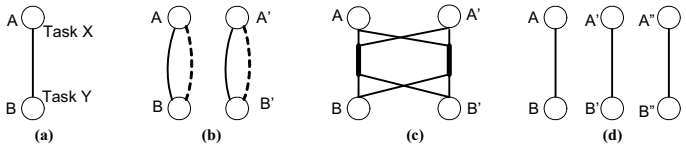

Fig.1. An illustration of the two-step mapping approach

Hereinafter, we will call the approach in (b) the cluster and path protection (CPP) and the approach in (d) the virtual network protection (VNP). In CPP, each logical connection c is protected from a link failure by establishing two disjoint paths and two copies of the job are provisioned to survive from any single cluster failure. In VNP, three disjoint copies of the jobs are provisioned to survive one link failure and one cluster failure. In general, suppose there are m connections and n tasks in the task graph, then we need a total of $2n$ clusters and $2m$ pairs of disjoint paths. In contrast, we need $3n$ clusters and $3m$ connections in VNP. Clearly, there is a trade-off between the network resources (e.g. wavelengths) and the computing resources (e.g. clusters). However, we may not simply conclude that CPP will need more wavelength resources because some logical connections in CPP (e.g. in Figure 1(b)) can share physical links (e.g. in Figure 1(c)). In general, different resource isolation constraints in these two approaches imply different complexities and success rates in mapping the ASAP topology onto the substrate network. Therefore, it is not clear which one of the proposed approaches will perform better and we will investigate them in the following.

3.2 Formulation

Based on the above models and assumptions, for a given job we need to map the ASAP topology onto the underlying WDM network by optimally 1) assigning tasks to computing clusters and 2) routing the lightpaths between clusters. For such an assignment to be feasible, the following constraints must be satisfied: 1) the capacity constraint on both links and clusters due to limited resources; 2) the flow conservation and demand satisfaction constraint; In addition to these common constraints in networking studies, we have another two problem-specific constraints: 3) the assignment and traffic dependency constraint, which reflects the fact that the source-destination of a connection c depends on the task assignment; 4) the resource isolation constraint. More specifically, for VNP, it ensures that the lightpaths belong to different ASAP networks of one job will not share any link and one task in different ASAP networks will not be assigned to the same computing cluster for execution. For CPP, two lightpaths associated with one connection are disjoint and two clusters executing the same task must be different.

3.3 An Algorithm for CPP

The proposed algorithm for CPP starts from an empty assignment, and iteratively completes a partial assignment by assigning some unassigned task to a compatible cluster that minimizes the total cost with respect to assignments already made. More specifically, for a task n and one of its compatible cluster v, the augmented cost with respect to

assignments already made consists of the execution cost ($\mathbf{e}_{nv}$) and the communication cost, which is not fixed and can be estimated by the following procedure. For an assigned tasks *n'* that is a neighbor of to *n*, 1) establish the primary path $P_{vv'}$(where *v'* is executing *n'*) on the shortest path that has sufficient bandwidth available with the cost $\mathbf{b}_{nn'}\mathbf{p}_{vv'}$; 2) set the cost of each link on the primary path is to $+\infty$ since the backup path needs to be disjoint with the primary path. For each link *e* (with shared bandwidth b_e^s and unassigned bandwidth b_e^u) along $P_{vv'}$, find all the existing primary paths of the same job over *e* and then their corresponding backup paths. Since two backup lightpaths can not share bandwidth on a link if their corresponding primary paths are not link disjoint, the cost of each link on those found backup paths is set to $\mathbf{b}_{nn'}$ if $b_e^u > \mathbf{b}_{nn'}$. Otherwise, set it to $+\infty$. For other links, shared bandwidth can be utilized for reserving the backup path. The link cost is then set to $\max(\mathbf{b}_{nn'} - b_e^s, 0)$ if $b_e^s + b_e^u > \mathbf{b}_{nn'}$. Otherwise, set the link cost to $+\infty$; 3) establish the backup path along the path with minimum cost using Dijkstra's algorithm based on the link costs decided on step 2. For an unassigned tasks n^* that is connected to *n* in the task graph, we just use the average path length to estimate the communication cost between n^* and *n*. After one ASAP network is provisioned, we remove all used clusters and find another ASAP network using the same algorithm described above.

3.4 An Algorithm for VNP

The proposed algorithm to find *K* (*K* =3 for two failures) disjoint ASAP networks upon a job's arrival is summarized below. For each $k < K$, we find the most cost-effective ASAP network (using an algorithm similar to the one in section 3.3) in the current network G_n and then remove it from G_n so as to guarantee all ASAP networks are disjoint. On one hand, the selected ASAP network should be "small" in terms of the number of links in it, which can increase the chance of finding other ASAP networks. On the other hand, for each ASAP network, its own objective is to minimize the cost. Therefore, we consider both objectives by using a convex combination of them as the new objective when deploying individual ASAP network.

4. Simulation Results

We have implemented various proposed heuristics over a WDM network with a USNET topology (24 nodes and 43 links). We generated 10,000 jobs which come and go according to a Poisson process (with average job arrival rate λ =10). Each job is represented by a 3-node graph, which is randomly generated as in [4]. Failures are simulated by randomly cutting a link or removing a computing cluster at the rate 0.001 λ. The performance of above proposed schemes is shown in Fig. 2. The results show that VNP has a lower job blocking rate (shown in the logarithm scale in (a)) than CPP when we assume sufficient computing resources (when each cluster has more than 25 CPUs) and varying link capacities. This is because VNP requires a fewer wavelengths for one job. However, with sufficient wavelength resources (when each link has more than 25 wavelengths) and varying cluster capacities as in (c), CPP has a lower job blocking rate than VNP because CPP requires a fewer CPUs for one job. In both cases of sufficient computing resources (b) or sufficient wavelength resources (d), VNP achieves a less average job cost than CPP under our assumption that the ratio of unit communication cost to the unit execution cost is 1. When both computing and wavelength resources are limited, it may not be feasible to separate the performance curves of CPP and VNP in a clear-cut way due to the complex supply/demand dynamics between computing resources, wavelength resources and input jobs. But basically the performance of CPP is less sensitive to the computing resources and more sensitive to the wavelength resources than VNP (related results are not shown here due to limited space).

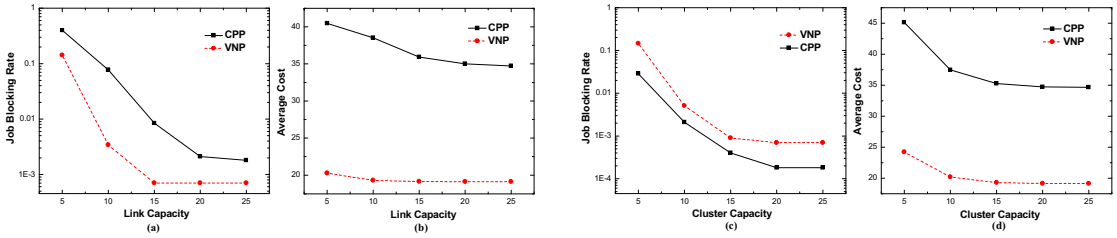

Fig.2. Job blocking rate vs. average job arrival rate

5. Concluding Remarks

Providing survivable ASAP networks in the presence of concurrent and multi-layer failures is critical for emerging applications under new paradigms such as SOA and Cloud Computing. Due to the major differences between an ASAP network and a virtual topology, new approaches to dedicated/shared protection of ASAP networks have been proposed, and new constraints, issues and tradeoffs identified.

6. References

[1] H. Choi, S. Subramaniam and H. Choi, "On double-link failure recovery in WDM optical networks," IEEE INFOCOM, **2**, 808-816, 2002.

[2] S. Kim and S. Lumetta, "Evaluation of Protection Reconfiguration for Multiple Failures in WDM Mesh Networks," OFC03, **1**, 210-211, 2003

[3] Kayi Lee and Eytan Modiano, "Cross-Layer Survivability in WDM Networks with Multiple Failures," OFC/NFOEC, OWN2, 2008.

[4] X. Liu, et al., "Survivable Optical Grids," OFC/NFOEC, OWN1, 2008.