One Node-Fault Tolerant Graph with Node's types for survivable Virtual Network Request with Service Type

Kun Xie[1], *Member, IEEE*,Heng Tao[1]
[1] College of Computer Science and Electronic Engineering,Hunan University, China

***Abstract*—As virtualization is becoming a promising way to support various emerging application, provisioning survivability to requested substrate networks (SN) which is embedded with virtual network in a resource efficient way is important. In this paper, First, we consider the failure dependent protection (FDP) in which each primary facility node would have a different backup facility node, as opposed to the Failure Independent Protection (FIP) which has been studied before, in order to provide the same degree of protection against a single node failure with less substrate resources**

In a virtualized infrastructure where physical resources are shared, a single physical server failure will terminate several virtual serves and crippling the virtual infrastructures which contained those virtual servers. To guarantee some level of reliability, each virtual infrastructure, at instantiation, should be augmented with backup virtual nodes and links.

I. Requesting Survival Virtual Network using Integer Linear Programming

The one of contributions of this paper is the modelization of specific node fault tolerant problem as an optimization. The general technique used to solve this problem is a combined algorithm consisting of enumeration and Ullmann Algorithm [1], which is proposed as a algorithm of exponential complexity which is brute force method briefly. More precisely, we propose an Integer Linear Program(ILP) [2] model of this problem. In the following, We briefly explain how to solve a problem using ILP.

A. Modeling the SNFT problem

In this section, we formulate the problems discussed in the previous sections as Integer Linear Program(ILP) problem, Though we have proved that these problems are NP-complete, as we will demonstrate later, the ILP formulation provides a very viable tool for solving it, for most real-world virtual network request which typically have less than a few hundred nodes and edges. Throughout this section, we will focus exclusively on the survival virtual network request $[G(V,E,S),B(V,S))]$, we assume that G is simple directed flag label graph.

To begin with, we introduce some necessary notation:

MBG: $[mbg_{u,v}]_{(|V|+|B|)\times(|V|+|B|)}$, the adjacency matrix of graph G^o. where

$$mbg_{u,v}=\begin{cases}1\ if\ edge\ e\ originates\ from\ u\ to\ v\\0\ otherwise\end{cases}$$

MAG: $[mag_{u,v}]_{|V|\times|V|}$, the adjacency matrix of arbitrary graph G. Where

$$mag_{u,v}=\begin{cases}1\ if\ edge\ e\ originates\ from\ u\ to\ v\\0\ otherwise\end{cases}$$

T^l: $[t^l_{u,v}]_{|V|\times(|V|+|B|)}$, The l-th node permutation matrix against l-th node failed. Where

$$t^l_{u,v}=\begin{cases}1\ if\ node\ u\ transformed\ to\ node\ v\\0\ otherwise\end{cases}.$$

MBS: $[mbs_{u,v}]_{(|V|+|B|)\times|S|}$, incidence matrix of specific flags of the SNFT graph G^o. where

$$mbs_{u,v}=\begin{cases}1\ if\ node\ u\ have\ specific\ flag\ v\\0\ otherwise\end{cases}.$$

MAS: $[mas_{u,v}]_{|V|\times|S|}$, incidence matrix of specific flags of the arbitrary graph G. where

$$mas_{u,v}=\begin{cases}1\ if\ node\ u\ have\ specific\ flag\ v\\0\ otherwise\end{cases}.$$

$AugB$:$[aug_i]_{|B|}$, whether the i-th backup nodes is used or not. where

$$aug_i=\begin{cases}1\ if\ node\ i\ is\ used\\0\ otherwise\end{cases}.$$

The problem of requesting 1-specific node fault tolerant graph $[G(V,E,S),B(V,S))]$ can be formulated as the folowing ILP problem:

$$such\ as:\ T^l*(T^l*MBG)^{'}\geq MAG \tag{1}$$

$$\sum_{1\leq v\leq(n+b)}T^l_{iv}=1(1\leq i\leq n) \tag{2}$$

$$\sum_{1\leq u\leq n}T^l_{uj}\leq 1(1\leq j\leq(n+b))$$

$$\sum_{1\leq u\leq n,1\leq v\leq(n+b)}T^l_{uv}=n$$

$$T^k_{uk}=0(1\leq u\leq n) \tag{3}$$

$$MAG_{uv}\leq MBG_{uv}(1\leq u\leq n,1\leq v\leq n) \tag{4}$$

$$MAS\leq T^l*MBS \tag{5}$$

$$\frac{\sum_{1\leq v\leq(n+b)}MBG_{(n+i)v}+n-1}{n}\leq B_i(1\leq i\leq b) \tag{6}$$

$$\tag{7}$$

- Constraint(1) guarantees that when the l-th critical node of graph G failed, we could find subgraph which is subgraph isomorphism of SNFT graph G^o. we represent the mapping relation as permutation matrix in view of Ullman [1]
- Constraint(3) implies that operation of the transform matrix of mapping relation is appropriate and correct format in respect with one node failure.

- Constraint(4) guarantees that SNFT graph G^o must be the augmented graph of previous graph G.
- Constraint(5) guarantees that when the l-th node failed all nodes of the transformed graph still have specific flag, which is specific flag of corresponding node of previous graph G.

B. Objective Function and Approximation

We seek to minimize the amount of resource used for a VInf. The object function of the adapted ILP is then

$$min : \alpha \sum_{1\leq i\leq b} B_i + \beta \times (\sum_{1\leq u\leq (n+b) 1\leq v\leq (n+b)} MBG_{uv} - \sum_{1\leq u\leq n, 1\leq v\leq n} MAG_{uv}) \quad (8)$$

where α and β are node and link weights, respectively. To achieve minimize the amount of nodes, the weight can be set as $(n^2 + 1)$ and 1, respectively.

The presence of the boolean variables turns the linear program into a NP-Hard problem. An alterative is to relax the boolean variables to real-value variables, obtain an approximate virtual node transformation.

$$dp[i][W_1][W_2]\dots[W_m] = \left\{ \begin{array}{l} dp[i-1][W_1 - w_i][W_2]\dots[W_m] + v_i \\ dp[i-1][W][W_2 - w_i]\dots[W_m] + v_i \\ dp[i-1][W][W_2]\dots[W_m - w_i] + v_i \end{array} \right\} \quad (9)$$

$n * \prod_{i=1}^{m} C_i$
$m * n * \prod_{i=1}^{m} C_i$

II. Evaluation

In the proposed two step for $EVSNR$, we employ a proactive failure dependent protection based EVN design with sufficient redundancy before embedding process in order to conquer the limitation of existing work, which consider the redundancy within the embedding process and failure independent protection based migration approach, in resource efficiency. Accordingly, the performance of optimal solution of EVN design and embedding formula, and heuristic algorithms are presented in this section in terms of average acceptance ratio, embedding cost, as well as the migration frequency after facility node failure. when a substrate node fail, which is placed with two virtual node.

A. Simulation Settings

For any VN request, the number of VN nodes is randomly determined by a uniform distribution between 4 and 10 and each pair of virtual nodes is randomly connected with probability 0.5. The computing requirements on VN nodes follow a uniform distribution from 5 to 10, as well as the bandwidth on VN links from 10 to 20. The arrivals of VN requests are modeled by a Poisson process (with mean of 10 requests per 100 time units).The duration of the requests follows an exponential distribution with 1000 time units on average. We assume the relative cost of computing and bandwidth is 3 [3], [4], which means $\alpha = 3$. The SN topologies used are randomly generated with 40 nodes using the GT-ITM tool [5] The computing (bandwidth) resources of the substrate nodes (links) are real numbers uniformly distributed between 30 and 50 (100 and 150).

B. Acceptance Ratio

In this section, the service acceptance ratio for VN with the proposed survivable approach is examined, as well as a typical integer liner program method. Also, the acceptance ratio of VN without survivability requirement is presented as a baseline to gauge the impact of additional amount of resources consumed for survivability on the service provisioning capability of SN.We employ the proposed $EVSNR$ as the embedding algorithm for all the EVN designing approach.

In Fig. 6, acceptance ratio drops quickly before 5000 time units because there are sufficient substrate resources for the arrived VN requests, and when the amount of running VN requests in system are stable (after 7500 time units), acceptance ratio would keep consistent. Fig. 6 also depicts that FDEVN: MinFir and FD-EVN:Rad lead to higher acceptance ratio than FI-EVN algorithm through efficient resources sharing. In particular, comparing with FI-EVN, FD-EVN:MinFir approach achieves almost 15long run. Intuitively, it is the result of fact that, for FI-EVN embedding, the backup node is associated with a lot of resources since it has to emulate every primary node after it fails, and so, embedding of backup node is vulnerable. However, the acceptance ratio suffers at least 8resources consumption for survivability purpose.

C. Embedding Cost

In this work, the embedding cost is calculated as the cost of the substrate resources (i.e. cost of CP. on all facility nodes and CM. on all fiber links) consumed to satisfy the EVN resource requirements. And, in this simulation, we assume there are sufficient resources in all the substrate components.

We further compare the embedding cost of different EVN design algorithms with (denoted as +Shared) or without (denoted as +NoShared) consideration of the substrate resources sharing in EVNE approach. Generally, with regular VN arriving and leaving, embedding cost.

D. Migration Frequency

As mentioned above, FD-EVN algorithms achieve higher acceptance ratio and lower embedding cost at the cost of more node migration after facility node failure, which would cause service interruption and should be examined carefully, especially for the application with SLA constraints. We run our simulation in 30000 time unites, which corresponds to about 2000 requests on average in each instance of simulation. The migration frequency after random facility node failure is presented in Fig. 8 in terms of the number of VN nodes.

The physical infrastructure consists of 40 compute nodes with capacity uniformly distributed between 50 and 100 units. These nodes are randomly connected with a probability of 0.4 occurring between any two nodes, and the bandwidth on each physical link is uniformly distributed between 50 and 100 units. VInf requests arrive randomly over a timespan of 800 time slots and the inter-arrival time is assumed to follow the Geometric

$$
\begin{array}{cc}
 & \begin{array}{cccccccccc} C_{P_{2.2}} & C_{P_{2.3}} & C_{P_3} & C_{P_4} & C_{B_{1.1}} & C_{B_{1.2}} & C_{B_{2.1}} & C_{B_{2.4}} & C_{B_{3.2}} & C_{B_{3.2}} \end{array} \\
\begin{array}{c} R_{V_1} \\ R_{V_2} \\ R_{V_3} \\ R_{V_4} \end{array} &
\left[\begin{array}{cccccccccc}
\infty & \infty & \infty & \infty & \boxed{\text{N+(2)+4+5+3}} & \infty & \text{N+(2)+4+5+3} & \infty & \infty & \infty \\
\boxed{4} & \infty & \infty & \infty & \infty & \text{N+(3)+4+6} & \infty & \infty & \text{N+(3)+4+6} & \infty \\
\infty & \text{M+(2)+5} & \boxed{5} & \infty & \infty & \infty & \infty & \infty & \infty & \text{N+(5)+5+6} \\
\infty & \infty & \infty & \boxed{3} & \infty & \infty & \infty & \text{N+(6)+3} & \infty & \infty
\end{array}\right]
\end{array}
$$

$$
\begin{array}{cc}
 & \begin{array}{ccccccccc} C_{P_1} & C_{P_3} & C_{P_4} & C_{B_{1.1}} & C_{B_{1.2}} & C_{B_{2.1}} & C_{B_{2.4}} & C_{B_{3.2}} & C_{B_{3.2}} \end{array} \\
\begin{array}{c} R_{V_1} \\ R_{V_2} \\ R_{V_3} \\ R_{V_4} \end{array} &
\left[\begin{array}{ccccccccc}
4 & \infty & \infty & \text{M+4} & \infty & \text{N+(2)+4+5+3} & \infty & \infty & \infty \\
\infty & \infty & \infty & \infty & \text{M+(1)+4+1} & \infty & \infty & \text{N+(3)+4+6} & \infty \\
\infty & 6 & \infty & \infty & \infty & \infty & \infty & \infty & \text{N+(5)+5+6} \\
\infty & \infty & 0 & \infty & \infty & \infty & \text{N+(6)+3} & \infty & \infty
\end{array}\right]
\end{array}
$$

$$
\begin{array}{cc}
 & \begin{array}{cccccccccc} C_{P_1} & C_{P_{2.2}} & C_{P_{2.3}} & C_{P_4} & C_{B_{1.1}} & C_{B_{1.2}} & C_{B_{2.1}} & C_{B_{2.4}} & C_{B_{3.2}} & C_{B_{3.3}} \end{array} \\
\begin{array}{c} R_{V_1} \\ R_{V_2} \\ R_{V_3} \\ R_{V_4} \end{array} &
\left[\begin{array}{cccccccccc}
\boxed{5} & \infty & \infty & \infty & M+5 & \infty & \text{N+(2)+4+5+3} & \infty & \infty & \infty \\
\infty & 6 & \infty & \infty & \infty & \boxed{\text{M+6}} & \infty & \infty & \text{N+(3)+4+6} & \infty \\
\infty & \infty & \boxed{\text{M+(2)+1}} & \infty & \infty & \infty & \infty & \infty & \infty & \text{N+(5)+5+6} \\
\infty & \infty & \infty & \boxed{0} & \infty & \infty & \infty & \text{N+(6)+3} & \infty & \infty
\end{array}\right]
\end{array}
$$

Accepted VNR Ratio	Ratio of virtual networks that were successfully embedded into the substrate topology.
Stress	Average number of virtual links/nodes that have been assigned to the substrate links/nodes
Utilization	Bandwidth/CPU utilization of substrate links/nodes.
Path Length	Length of communication paths assigned to virtual links.
Cost, Revenue, and Cost/Revenue	Cost: Sum of CPU and bandwidth resources being used for the embedding. Revenue: Sum of CPU and bandwidth demands realized
Active Nodes	Number of nodes that need to be active in order to host all the virtual networks. This metric is especially useful in the context of e
Power Consumption	Power consumption of all Active nodes. Several powerconsumption models are implemented.
Runtime	Average runtime of the algorithm.
Initialization Overhead	Some algorithms come with initialization cost (in terms of runtime), e.g., the distributed algorithm presented in initially partitions th
Hidden Hops Ratio	Ratio of hidden hops, e.g., the number of nodes only needed for forwarding packages between other nodes. Especially useful in the
Communication Overhead	Communication overhead of distributed algorithms (i.e., number of messages sent between substrate nodes).

distribution at a rate of 0.75 per time slot. The resource lease times of each VInf follows the Geometric distribution as well at a rate of 0.01 per time slot. A high request rate and long lease times ensures that the physical infrastructure is operating at high utilization. Each VInf consists of nodes between 2 to 10, with a compute capacity demand of 5 to 20 per node. Up to 90% of these nodes are critical and all failures are independent with probability 0.01. Connectivity between any two nodes in the VInf is random with probability 0.4, and the minimum bandwidth on any virtual link is 10 units. There are two main sets of results: (i) scaling the maximum bandwidth of a virtual link from 20 to 40 units while reliability guarantee of every VInf is 99.99%, and (ii) scaling the reliability guarantee of each VInf from 99.5% to 99.995% while the maximum bandwidth of a virtual link is 30 units.

Accept ratio mapping cost migration cost cpu time per Vinf

References

[1] J. R. Ullmann, "An algorithm for subgraph isomorphism," *Journal of the ACM (JACM)*, vol. 23, no. 1, pp. 31–42, 1976.

[2] A. Schrijver, *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, *et al.*, "Above the clouds: A berkeley view of cloud computing," tech. rep., Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, 2009.

[4] H. Yu, C. Qiao, V. Anand, X. Liu, H. Di, and G. Sun, "Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–6, IEEE, 2010.

[5] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to model an internetwork," in *INFOCOM'96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, vol. 2, pp. 594–602, IEEE, 1996.