VIRTUAL LINK CONFIGURATION FOR THE BACKBONE AUGMENTATION IN AN INTERNET LINK-STATE ROUTING DOMAIN

Dr. Dohoon Kim, Kyung Hee University, dyohaan@khu.ac.kr

ABSTRACT

This paper addresses the issue of augmenting backbone configuration, which arises from ISP's hierarchical Link-State(LS) routing operations. Focusing on Virtual Link(VL) configuration, proposed is an efficient augmentation scheme of increasing redundancy in the original backbone topology. A VL restores hidden information in the topological database of each backbone router, thereby increasing redundancy in the backbone and preventing partition of the backbone when some links fail. With given potential VL locations and the original backbone topology, we propose a bi-objective optimal VL placement model together with a branch-and-bound algorithm, which explicitly evaluates the benefit side as well as the cost side of VL configuration, and fully leverages the trade-off between both sides. To our knowledge, this is the first try to attack systematically the VL configuration problem in hierarchical LS routing. The proposed model and algorithm are expected to relieve network administrators from the burden of configuring VLs, support making the backbone more tolerable to backbone link failures, and finally, provide a practical vehicle for reliable LS hierarchical routing indispensable to overall service quality.

Keywords: Link-state routing, OSPF, Virtual link, Multi-objective, Branch-and-bound algorithm

INTRODUCTION

As the number of Internet users grows, scalability issues for example, a routing information overflow, are emerging as one of the most fundamental network operation problems since the competitive edge of the ISPs is built around routing efficiency. To cope with scalability issues, ISPs hierarchically divide their own networks(so-called Autonomous Systems(ASs)) into two tiers: local distribution areas and a single backbone. As a result, ISPs can limit the range of routing information exchange and resolve the major scalability issue. However, this gain from hierarchy comes with some side effects. Besides operational inconvenience, the largest cost of hierarchical configuration is the performance degradation due to limitation of available routes. Furthermore, in hierarchical implementation of link-state routing protocols, such as OSPF(Open Shortest Path First) and IS-IS(Intermediate System-Intermediate System), the overall performance of the entire AS comes to depend on the backbone since the routers in the backbone play the core functions to collect, organize, and redistribute routing information across local distribution areas. Therefore, sound connectivity of the backbone is a critical success factor for running hierarchical LS(Link-State) networks([1], [8], [10], [11]).

Along this line, an efficient method to increase redundancy of the backbone should be developed as one of the measures to make the backbone robust. Redundancy is important to prevent the backbone from being disconnected when backbone links or nodes fail. For example, two edge-connectivity should be satisfied in every pair of backbone routers by configuring back-up links so that no single link or node failure may cause disconnection of the backbone.

Fortunately, many link-state protocols provide an easier mechanism to generate back-up links than constructing new links incurring high cost. Remark that with hierarchical link-state routing

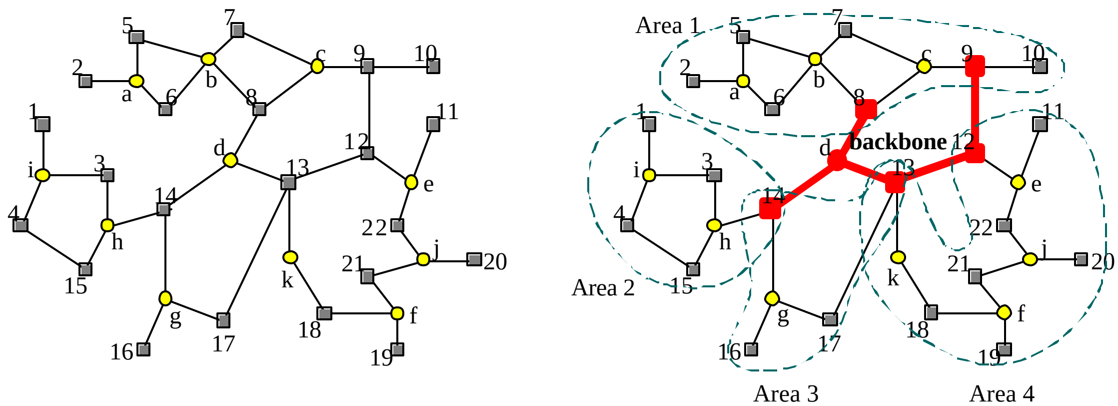
protocols, every backbone router maintains a map(so-called Topological DataBase, TDB) representing the physical backbone topology and may have alternative routes that are not present in its TDB unless the entire AS is disconnected. Virtual Links(VLs) restore such hidden information in the TDB of the backbone routers. Since little monetary expense and time is required to configure VLs, they are recommended as the basic option to introduce redundancy into the backbone with leaving the operational flexibility intact.
However, many network administrators still find it difficult to decide where to configure VLs because of possible inconvenience and operational complexity caused by misplaced VLs. Except some practical tips, few literatures deal with an efficient method or even a rule-of-thumb to configure VLs in the backbone routers in order to enhance the backbone connectivity([10]). We prospect that VL configuration issues will come to the surface as link-state protocols are attaining the status of de-facto standard for Internet routing. To our knowledge, this paper is the first attempt to attack the VL configuration problem in a well-structured manner.
In the next section, we first describe the VL back-up system and then analyze the costs and benefits arising from configuring VLs. Section 3 provides a mathematical model for determining optimal locations of VLs in terms of both benefits and costs. We present a solution method based on branch-and-bound algorithm and some numerical examples in section 4. By summarizing important features of the model and the results, and presenting future works, we conclude this paper.

VIRTUAL LINK CONFIGURATION MODEL

Roles of Virtual Links in Hierarchical Link-State Networks

In LS routing protocol, routers exchange their piecemeal topological information and construct a map(TDB) representing the overall network topology. As the network size increases, the volume of these transactions and TDB size grow exponentially. So does the resource consumption for routing management and control. Because of this scalability issues, large LS networks are partitioned into relatively small local areas within each of which routing information exchange among member routers is limited. In order to maintain the entire connectivity(i.e., inter-area routing), a backbone connecting local areas should be constructed. That is, a valid backbone must be contiguous(i.e., all backbone routers should be connected to other backbone routers through backbone links only) and ensure direct connection of each local area ([8], [10]). Figure 1 shows an example of a valid hierarchical configuration.
Therefore, backbone is extremely important in hierarchically configured LS network. Good practices for backbone design suggest that stability and redundancy are the most important criteria for the good backbone([1], [8], [10], [11]). After hierarchy design, enhancing the backbone configuration in terms of connectivity should be accomplished. This augmentation, which can be realized by introducing redundancy with VLs, aims at preventing partition of the backbone when some links fail. Accepted LS network design practices indicate that the use of VLs should be considered for a backbone poorly designed as a result of unbalanced hierarchical configuration([4], [8], [10]).
A VL can be configured between separate gate routers that touch the backbone from each side and have a common area. In the case of figure 1-(b), at least three VLs, which connect routers 8 and 9, 12 and 13, 13 and 14 across area 1, area 4, and area 3, respectively, can be configured. Then the VL acts in a similar way as in a tunnel: i.e., a VL creates a path between two gate nodes

by using non-backbone links. A VL can only be configured on gate nodes, and in general, only one of the potential VL configurations passing the same local area is configured. The stability of a VL is dependent on the stability of the area it traverses, and the reliability measure(cost metric) of a VL can be defined as that of the weakest link of the links constituting the VL. Also, the amount of the effort required to configure and maintain a VL is proportional to its length(the number of routers on the path corresponding to the VL).



(a) original AS topology (b) hierarchical clustering of AS

Figure 1: Hierarchically Clustered LS Network and VL Configuration Example

Even though it is desirable to configure sufficient VLs to build back-up links, additional VL configurations required for this purpose put lots of burden on the backbone and increase the complexity of the system operation due to the characteristics of the VLs. Therefore, configuring all the potential VLs may lead to an inefficient augmentation of the backbone connectivity. In the following sections, we propose an efficient VL configuration model which fully leverages the trade-off between the benefits and costs of configuring VLs and solution methods to this model.

A Decision Model for VL Configuration

The survivable network design approaches and solution methods([3]) are no longer applicable to the most practical situations where we cannot assume that connectivity requirements of backbone routers are given or available VLs cannot fulfill the given requirements. For these situations where connectivity requirements have no meaning, the single objective decision model, such as minimizing total cost, should be extended in order to explicitly evaluate the benefits and costs of configuring VLs and to achieve an optimal balance between benefits and costs. Accordingly, the issue of configuring VLs should be multi-objective decision model looking for an augmented backbone topology which maximizes the net gain(= total benefit – total costs) by setting-up VLs. The decision problem in this paper is to determine an optimal configuration of VLs with a given set of potential VL locations and the original backbone topology. In this decision context, a decision alternative is an augmented graph from the original backbone topology by choosing some VLs(decision variables). Thus, in general, there exist finite but many decision alternatives. We will evaluate a decision alternative in terms of two attributes: the connectivity level of the augmented graph and the cost for configuring VLs in the augmented graph. Specifically, compensatory preference(so-called utility function of the attributes) approach will be employed for this bi-criteria decision making problem in order to explicitly reflect the trade-off between attributes and to resolve commensurability issues for comparing heterogeneous attributes. In this paper, we impose a minimum level of restrictions on the form of utility functions so that the network administrator can have sufficient flexibility to develop unique utility functions customized to their own situations.

We are given the original topology of the backbone that can be represented as a simple graph $G = (V, E)$ where V and E represent the set of backbone routers and the set of backbone links, respectively. Also, given is A, the pre-defined set of potential VL locations (the set of decision variables) with assigned installation cost w_e ($e \in A$) which is proportional to the length of VL e. Let $G_Q = G(V, E \cup Q)$ denote an augmented topology by $Q(\subseteq A)$, which corresponds to a decision alternative. Note that G_Q may not be a simple graph any longer. The cost vector of configuring VLs in Q is defined as $\tilde{w} = (w_1, \ldots, w_{|Q|})$. And the nodal connectivity c_v on G means $\min_{u \in V \setminus \{v\}} \gamma(v, u \mid G)$, where $\gamma(v, u \mid G)$ is the maximum number of link-disjoint paths between u and v on G. That is, for a given node v, any failure of $(c_v - 1)$ links cannot cause node v isolated from the other part of the backbone. According to this definition, all the nodal connectivity has the same value corresponding to a bottleneck point of the given graph G. This value is, in fact, the cardinality of a minimum cut of G and will be written as $\bar{c}$. Gomory-Hu tree method and other variants ([5], [9]) provide efficient mechanisms to compute $\bar{c}$. After determining Q from A, $\bar{c}$ and $\tilde{w}$ represent the attributes of an alternative G_Q.

To complete the decision model for optimal configuration of VLs, utility functions for attributes are employed. First, we introduce $U_C(G_Q)$ and $\bar{U}_W(G_Q)$, each of which represents the network administrator's normalized evaluations of benefit from connectivity level $\bar{c}$ of G_Q and cost to install and maintain $|Q|$ VLs, respectively. $\bar{U}_W(\text{-})$ can be interpreted as disutility of costs incurred by VL configuration, and is assumed to be $\bar{U}_W(G_Q) = \sum_{e \in Q} w_e / \Delta$ where $\Delta = \sum_{e \in A} w_e$. Both $U_C(\text{-})$ and $\bar{U}_W(\text{-})$ are monotonic non-decreasing function of the connectivity level and the sum of lengths of VLs in Q, respectively. Finally, the network administrator's total utility is described as an additive form of $U(G_Q) = \alpha\, U_C(G_Q) - \beta\, \bar{U}_W(G_Q)$ where α and β are weight coefficients. Figure 2 skeletonizes the decision model for Optimal VL Configuration [OVLC].

$$\textbf{[OVLC]} \qquad \text{Max}_{Q \in A}\; U(G_Q)$$

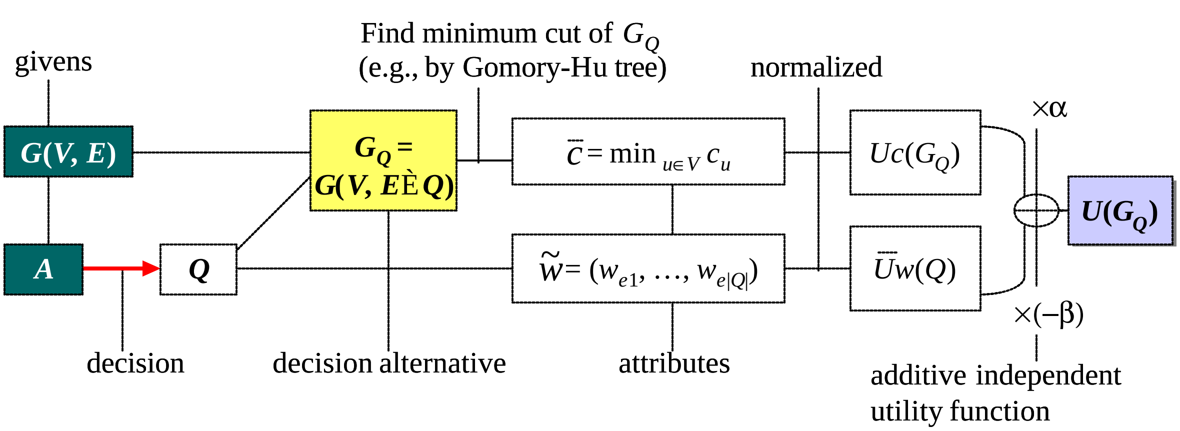


Figure 2: Decision model of Optimal VL configuration

SOLUTION METHOD AND EXAMPLES

Branch and Bound Algorithm

[6] demonstrates that [OVLC] is not an easy problem by presenting a counter example that shows a naive solution method based on component-by-component comparisons of decision variables could not succeed. This example implies that we may have no choice but to examine $2^{|A|}$ decision alternatives to find optimal solution in the worst case. However, considering the typical size of the ISP's backbone and the number of potential VL locations, enumerating all the alternatives discourages practical implementation of [OVLC] model. At this point, an efficient enumeration method is requested. We present a branch-and-bound algorithm(implicit enumeration) with good lower and upper bounds so that solution speed can be accelerated. Each [RP^k], Restricted Problem k in the branch-and-bound tree is characterized as a partition of $\boldsymbol{A}$(set of potential VL locations), $\{A^k, \overline{A}^k, F^k\}$ where A^k and $\overline{A}^k$ represent the sets of VLs that should be included and excluded in [RP^k], respectively, and $F^k = A-(A^k \cup \overline{A}^k)$. Let a relaxation problem of [RP^k] be denoted by [R-RP^k]. Employed is an objective relaxation to compute upper bound to the [RP^k] easily; that is, ignoring cost attribute(i.e., dropping the term $\overline{U}_W(G_Q)$ from the objective) defines [R-RP^k]. Since $U_C(G_Q)$ is a non-decreasing function of the connectivity level of G_Q and it is also non-decreasing to the density of the augmented graph, an optimal solution of [R-RP^k] is $Q_R{}^k = A^k \cup F^k$ with its value of $z_R^k = U_C(G_{Q_R^k})$ as an upper bound to [RP^k]. Also, let $z(Q_R{}^k) = U(G_{Q_R^k})$ then $z(Q_R{}^k)$ becomes a trivial lower bound to [RP^k].
However, the following property gives a way to construct a tighter lower bound from $Q_R{}^k$. The property can be also used to choose branching variables or to fix some variables in F^k during exploration of an optimal solution to [RP^k], thereby reducing the search space of the branch-and-bound algorithm. Before presenting the property, we define $L(e|G_Q) = U_C(G_Q) - U_C(G_{Q-\{e\}})$, which denotes the benefit loss from dropping a VL e from Q.

Property: If the following inequality (1) holds for an element ε in F^k of [RP^k], the subset $Q^\circ = A^k \cup (F^k - \{\varepsilon\})$ cannot be an optimal solution to [RP^k].

$$L(\varepsilon \,|G_{Q_R^k}) > w_\varepsilon \quad \ldots\ldots (1)$$

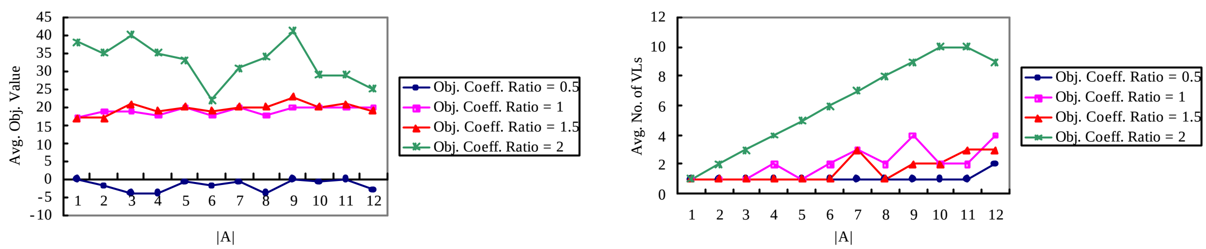
Proof) Refer to [6].

Inequality (1) means that cost-saving achieved by uninstalling the VL ε is large enough to offset the utility degradation due to the possible connectivity losses caused by losing ε on $G_{Q_R^k}$. Let's denote $g(e|G_Q) = w_e - L(e|G_Q)$ as an expected net gain from keeping VL e on G_Q. We can use this property to estimate an upper bound to a successor of [RP^k] when dividing the [RP^k] with ε as a branching variable: that is, as an upper bound to a child problem [RP^{k+1}] defined by excluding ε in consideration(i.e., $\overline{A}^{k+1} = \overline{A}^k \cup \{\varepsilon\}$ and $F^{k+1} = F^k - \{\varepsilon\}$). Since the optimal value of [R-RP^{k+1}] is already known to be worse than that of [R-RP^k], it is likely to terminate the branch-and-bound process early if we defer choosing [RP^k] as the next exploring candidate. In sum, the results from the property will guide branching variables as well as choosing restricted problems in the enumeration process.
We could also build a solution(let's say Θ) which consists of only VLs satisfying the inequality (1) at each [RP^k], thereafter comparing the total utility of Θ with that of $Q_R{}^k$. Since those VLs

seem the most profitable VLs on $G_{Q_R^k}$, we can expect that Θ produces better lower bound than Q_R^k. Even if Θ is no better than Q_R^k, we can improve Θ by sequentially adding the remaining free VLs in the decreasing order of $g(e|G_{Q_R^k})$, hoping that a better solution could be achieved in the course of sequential addition. Detailed description of our branch-and-bound algorithm can be found in [6].

Experimental Results

We have tested the proposed decision model and branch-and-bound algorithm with some randomly generated instances. The costs of implementing/maintaining VLs and the potential locations as well as the available number of VLs are randomly chosen within some ranges. For utility of nodal connectivity and disutility of VL cost, employed are a concave logarithmic function and a linear function, respectively. Since many literatures on decision analysis(for example, [2]) provide a lot of utility functions used in practical situations, a network administrator facing the decision problem [OVLC] can consult these literatures to choose the best one for his/her own purpose. Lastly, given functional forms of $U_C(\text{-})$ and $\overline{U}_W(\text{-})$, we tried various combinations of weight coefficients in the total utility function $U(\text{-})$ to see how network administrator's relative preference on the attributes affects the experiment results.



(a) Changes in the total utility values (b) Changes in the no. of VLs in optimal solutions

Figure 3: Simulation Results

varying the size of VL locations(|A| = 1, …, 12) and weight ratio(α/β = 0.5, 1.0, 1.5, 2.0)

Figure 3 summarizes sensitivity analysis results by varying the weight ratio(α/β) and the size of potential VL locations(|*A*|). Note that for each instance of varying size of *A*, we randomly generated ten different sets of potential VL locations in order to see the average behavior of optimal solutions of [OVLC]. As these graphs indicate, the amount of available resources(|*A*|) seems to differently affect the optimal solutions(the optimal structures of alternative decision, G_Q^*) and the optimal total utility values. The optimal utility values are rather independent of |*A*|, whereas the number of VLs in optimal solutions tends to rise as |*A*| increases. However, impacts of utility weight ratio on the number of optimal VLs as well as the optimal total utility look very strong. Furthermore, the result of experiments implies that a certain range of utility weight ratios lead to similar behaviors: for example, the graphs corresponding to the weight ratio 1 and 1.5 coincide at many levels of |*A*|.

CONCLUSION

In this paper, we proposed a framework for enhancing the backbone configuration of ISPs' hierarchical LS AS. Good backbone configuration should be robust so that some link failures may not cause an isolation of a portion of the backbone. Focusing on Virtual Link(VL) installation, proposed is an efficient augmentation scheme of increasing redundancy of the original backbone topology. With given potential VL locations and the original backbone topology, we propose a bi-objective efficient VL configuration model which explicitly evaluates the benefit side as well as the cost side of VL installation, and fully leverages the trade-off between both sides. Furthermore, provided is an efficient branch-and-bound algorithm with good lower and upper bounds so that solution speed can be accelerated in most cases.
To our knowledge, this is the first try to systematically address the VL configuration problem. For an ISP with its own domain, the proposed model and algorithm are expected to relieve network administrators from the burden of configuring VLs, support making the backbone more tolerable to backbone link failures, and ultimately provide a practical vehicle for reliable link-state hierarchical routing indispensable to overall service quality. For example, the framework can be employed to solve short-term network connectivity issues.
Due to unavailability of studies of the same kind, we could not present experiment results with real data. However, in the following research, comprehensive experiments will be conducted with several kinds of underlying backbone networks which are randomly generated but devised to reflect the real ISP network characteristics(for example, refer to [12]). Different forms of utility functions should be also tested in order to provide and analyze options that a network administrator may take in practice. In particular, we are developing a set of normalized multi-attribute utility functions which are not compulsory but can be employed in the common practice of multi-criteria decision models([2]).

REFERENCES

1. Behrens, J. and J.J. Garcia-Luna-Aceves (1998). Hierarchical Routing Using Link Vectors, Proceedings of IEEE INFOCOM 98, (2), 702-710.
2. Clemen, R.T. (1996). Making Hard Decisions: an Introduction to Decision Analysis, 2nd ed., Duxbury Press.
3. Grotschel, M., C.L. Monma, and M. Stoer (1995). Design of Survivable Networks, in Network Models, eds. M.O. Ball et al., North-Holland.
4. Halabi, S. (1996). OSPF Design Guide, Cisco Systems White Paper.
5. Hu, T.C. (1982). Combinatorial Algorithms, Addison Wesley.
6. Kim, D. (2002). Virtual Link Operations System for a Hierarchical Link-State Routing Domain, Working Draft.
7. Kim, D. and D.W. Tcha (2000). Scalable Domain Partitioning in Internet OSPF Routing, Telecommunication Systems, (15), 113-128.
8. Moy, J.T. (1998). OSPF: Anatomy of an Internet Routing Protocol, Addison Wesley.
9. Padberg, M. and G. Rinaldi (1990). An Efficient algorithm for the Minimum Capacity Cut Problem, Mathematical Programming, (47), 19-36.
10. Thomas, M.T. II. (1998). OSPF Network Design Solutions, Cisco Press.
11. Tsai, W.T., C.V. Ramamoorthy, W.K. Tsai, and O. Nishiguchi (1989). An Adaptive Hierarchical Routing Protocol, IEEE Transaction on Computers, (38), 1059-1075.
12. Zegura, E.W., K.L. Calvert, and S. Bhattacharjee (1996). How to Model an Internetwork, Proceedings of IEEE INFOCOM 96, (3), 594-602.