A Heuristic Method for Hierarchical Partitioning of Link-state Internet Routing Domain

D. Kim[†], J.H. Kim, and D.W. Tcha[‡]

Graduate School of Management, KAIST(Korea Advanced Institute of Science and Technology)
207–43, Cheongryangri-Dong, Dongdaemun-Gu, Seoul, 130–012, Korea

ABSTRACT

Hierarchical routing mechanisms based on hierarchical partitioning of Internet routing domain, so-called AS (Autonomous System), have been proposed to cope with scalability problems. For ISPs (Internet Service Providers) running link-state routing protocol, it is critical to secure Internet connection to reduce the routing control overhead (for example, synchronizing routing database over routers, etc.) by hierarchically partitioning the AS. However, there are few literatures addressing hierarchical partitioning schemes for link-state Internet routing protocols such as OSPF. Link-state routing protocols require an additional constraint of contiguity for each resulting partitioned area: topological constraint. Also, a subnetwork constituting a AS should not be separated into different partitioned areas. Our goal is to develop a HC(Hierarchical Configuration) method under these constraints by determining the backbone boundary. To achieve this goal, we first devise a conceptual and graphic models of a AS. Then, we formulate the HC issue upon the graph as a combinatorial problem with new additional constraints. The objective is to minimize the backbone size which is crucial to overall stable operation. Finally, we present a heuristic solution method and experiment results.

KEYWORDS

Internet, Link-state Routing, Hierarchical Configuration, Heuristic, Scalability

1. Introduction

[†] Presenter

[‡] Corresponding author: tchadw@sorak.kaist.ac.kr (with CC to dhkim@kgsm.kaist.ac.kr)

As the number of Internet users grows, scalability issues are emerging as one of the most challenging network operation problems facing ISPs whose competitive edges are built around routing efficiency. Scalability issues exist at every layer of the Internet architecture: at IP(Internet Protocol) routing layer, for example, the explosion of flooded routing information is witnessed as the network size increases. The common practice of over-engineering neither is an economic approach at least regarding scalable routing problem nor accommodate ever-increasing traffic throughout the network. Considering today's competitive network service market, more economic and flexible measures to cope with the routing scalability are required. For economic management of that kind of scalability, ISPs hierarchically divide their own networks(so-called Autonomous Systems(ASs)) into two tiers: local distribution areas and the backbone, thereby limiting the range of routing information exchange and resolving the major scalability issue.

In this paper, we concentrate on the issues arising from Hierarchical Configuration(HC) of link-state routing protocols, such as OSPF(Open Shortest Path First) and IS-IS(Intermediate System-Intermediate System). It is link-state routing protocols that have been known suitable for hierarchically structured routing environment. Furthermore, many research results indicate that thanks to its ability to explicitly deals with topological information, the link-state routing is gaining superior position over other routing protocols for QoS(Quality of Service) environment, thereby becoming prevalent. Therefore, we should develop frameworks to explicitly take into account new constraints arising from protocol-specific characteristics.

Recently, best practices for running hierarchical link-state routing have been introduced(for example, [1] and [2]). In these literatures, identified are some unique technical constraints for valid HC as well as performance constraints for scalability. However, few literatures systematically develop such arguments to the extent of mathematical and computerized framework for how to make an effective HC under these constraints. Our goal is to develop a method that provides valid link-state HC satisfying all the constraints while minimizing risk of unstable operation. The proposed scheme places no restriction on the structure of the network topology. The model and design guidelines highlighted in the following sections provide a foundation on which ISPs can construct a reliable, scalable link-state network.

In the next section, we briefly discuss the hierarchical link-state routing protocol and define the problem of effective HC. Section 3 presents the mathematical model with a conceptual and graphic model of an AS and heuristic solution methods displayed by a simple example. By summarizing the presented approach and discussing the future works, we complete this paper.

2. Hierarchical Configuration of a Link-state Routing Domain

2.1 Link-state Routing and Hierarchical Configuration

In the link-state routing, also known as the 'topology technique([3])', each router keeps track of the explicit topology of the entire network by repeatedly flooding the information of incident link-states via data frames called Link-State Advertisements(LSAs). The LSAs gathered at a router collectively form a LSA DB which contains the full network information such as topology, link costs, etc. In the OSPF arena, an LSA DB is called a Topological DB(TDB). Routes are then determined on-demand fashion by running Dijkstra's shortest path algorithm with the latest TDB. The complexity of synchronizing routing information(i.e., TDB) between routers and of processing information at routers increases as the network grows, posing the scalability problem. In order to alleviate this problem, required is a HC which reduces routing overhead.

HC makes a routing domain logically divided, or partitioned, into a collection of subnet groups. Such a subnet group with member routers, each of which has an interface to a subnet in the group, forms an area. A link-state routing domain is then partitioned into one backbone and multiple Local Areas(LAs) and, upon partitioning, can be viewed as a two-level hierarchy with the specially designated backbone at the top level. The border of an area is composed of only routers, each of which will be simply referred to as a gate router. The constituents of the backbone are subnets, routers, and links that are not included in any LA, and all gate nodes. Furthermore, the backbone should be connected in its TDB configuration for the entire network connectivity.

After HC, all routers are not with the same TDB. A gate router, unlike non-gate routers, keeps a separate TDB for each of the areas that it belongs to. However, every router in the same area still has the same shape of TDB: therefore, the common TDB will be referred to as the TDB of the area. Gate routers condense routing information into summary links, thereby, a usual router has only partial information outside its own area, which reduces the usage of router resources such as memory and CPU time. HC further reduces the bandwidth requirement for synchronizing routing information since the scope of routing information as well as its flooding range are restricted within an area.

The gain from hierarchy gives rise to some side effects. Recall that both intra and inter-area routing decisions are made solely based on the links in its own area, inevitably causing some degradation of overall route quality. In the worst case, the quality of some route may be $O(n)$ times worse than that of the optimal route. However, in practice, the level of degradation in route quality is found not so significant as to offset the benefit from scalability. This, as in the previous studies ([3], [4], etc.), allows us to consider the domain partitioning without paying attention to the quality of routing. However, the overall performance of the entire AS heavily depends on the backbone since the routers in the backbone play the core functions to gather, organize, and redistribute routing information across local distribution areas. Network administrators should configure hierarchy in a way to minimize potential factors causing the performance degradation.

2.2 Hierarchical Configuration Decision of a Link-state Network

2.2.1 Decision Factors for LSHC

The capability to properly scale a link-state network is determined by a multiple factors: router memory requirements, CPU cycles, available bandwidth, and so on. Thus, the hierarchy does not need to be structured to provide good routes, but only has to provide basic connectivity while using small amount of resources. Many simulation results indicate that among these bottleneck resources, the most critical factor for scalability is CPU cycles in a router([5], [6], [7], etc.). Accepting this fact, we focus on router's CPU usage, which shows sharp increase according to the size of the area where the router resides.

The following is the conditions required for HC.

Technical conditions:

- Subnet should not be splitted over different areas. ......(1)
- LAs should be contiguous. ......(2)

These conditions are peculiar to link-state routing protocols.

Performance conditions 1:

- The number of routers in a LA is less than or equal to the pre-defined bound (LA size constraint).......(3)
- The number of LAs connected to a gate router. ......(4)

These conditions are required for stable system operation of the link-state protocol due to its special characteristics: that is, link-state routing protocol uses the CPU-intensive SPF algorithm Experience has shown that 40 to 50 routers per LA should be an upper limit([1], [8], [9]). Also, it would be better not to overload a gate router. Thus, in general, one router should not be in more than three areas including the backbone([1], [2], [9]).

Performance conditions 2:

- A contiguous backbone must be present. ......(5)
- All LAs must have a direct connection to the backbone. ......(6)

Even though this category of conditions does not necessarily constitute a mandatory rule, many practices recommend these conditions be satisfied since they provide a good structure to guarantee the entire AS connectivity([1], [2], [9]).

2.2.2 Guidelines from the Best Practices

The goal of HC is to minimize the chance of performance degradation from hierarchy while satisfying all the conditions described in 2.2.1. The best practices have shown that ensuring the stability of the backbone is crucial to the overall AS performance, recommending a small and simple backbone([1], [9]). And the size of the backbone is defined in the same way as that of LA: the number of routers in the backbone. Therefore, the fewer routers are in the backbone the better overall performance do we expect.

However, the conditions in 2.2.1 are likely to result in a HC with small, many LAs, which implies many gate nodes, and in turn, increases the backbone size([4], [10]). On the contrary, in terms of the small backbone objective, LAs tend to be large enough to minimize the number of the backbone routers since the link-state protocol demands that all gate routers be backbone routers. The designer should balance this trade-off: the size of the LA/the number of the LAs and the backbone size.

Now we pose the decision problem of Link-State Hierarchical Configuration(LSHC) as determining the minimum backbone configuration on the given AS under the conditions (1) to (6). In the next section, we will see that some constraints make LSHC easy: for example, after choosing the backbone under the conditions (1) and (5), conditions (2) and (6) enforce the possible LA configurations confined in the resulting component structure on the complementary network.

3. Model and Solution Methods

3.1 Network Model

We start this section by introducing a new network model customized to describe the AS running link-state routing correctly. Distinguishing the roles of routers and subnets, an AS is described as an undirected graph $G=(V, E)$, where $V=V_1 \cup V_2$ $(V_1 \cap V_2 = \emptyset)$. V_1 and V_2 are the sets of router nodes and subnet nodes, respectively. A link in E represents a subnet-to-router or router-to-router interface. Note that classic Internet models in the literature do not distinguish differences between the two types of nodes, only showing the logical connections between routers([11]). It is this distinction that allows our network model to properly address the technical requirement associated with HC: for example, condition (1) now explicitly states that all the links incident to the same subnet node should belong to the same LA or the backbone.

3.2 Hierarchical Configuration Model for Link-state Network

To address [LSHC] problem mathematically, we may need the following notations and definitions. Suppose that S is a non-empty subset of nodes. Let $G[S]$ be a subgraph of G induced by node subset S. The induced subgraph $G[V-S]$ is denoted by simply $G-S$, the subgraph obtained from G by deleting the nodes in S together with their incident links. For our purpose, the size of a graph G, denoted by $|G|$, is defined as the number of type 1 nodes(router nodes) in G. When we are given a subgraph G_B of G, a gate node is the node who belongs to G_B and has an incident link that does not belong to G_B. Also, a subgraph G_B is said valid if all the incident links of a type 2 node(subnet node) in G_B also belong to G_B: therefore, the boundary of a valid subgraph G_B is composed of only type 1 nodes. Finally, given G with a special node subset S, an admissible union is a spanning family of connected subgraphs each of which does not share links and the remaining nodes other than S with other members of the family.

With the network model in 3.1 and L(given upper bound on the LA size), LSHC can be viewed as the following combinatorial optimization problem:

[LSHC]

Choose the minimal connected valid subgraph of G, $G_B = (V_B, E_B)$ whose gate node set is V_S, so that each resulting component C_i on $\overline{G}_B = G - (G_B - V_S)$ satisfies the following feasibility condition.

Feasibility condition:

$C_i = (V_i, E_i)$ is a connected subgraph of G with $V_i^S = V_i \cap V_S$ as its gate node set. And there exists an admissible union of $\{B_i^1, \ldots, B_i^m\}$ such that $|B_i^k| \leq L$ for $k = 1, \ldots, m$ and $|\{ k: v \in V_i^S, v \in B_i^k \}| \leq 2$ (that is, a gate node cannot be overlapped more than twice).

For a given component resulting from any potential backbone profile, the feasibility condition checks whether a feasible LA configuration is possible on the component or not. Validity and connectedness of the backbone configuration together with the feasibility on each resulting component in [LSHC] guarantee that the conditions for LA configuration((1) to (4) and (6)) will be satisfied, thereby arriving at a HC. Also, we can easily see that the following property and corollary hold.

Property 1

Suppose that a component of G, $C_i = (V_i, E_i)$ with V_i^S as its gate node set is given. If M, a subset of V_S^i is a node-cut, the component is an admissible union of more than two overlapping subgraphs, $B_i^1, \ldots, B_i^m$ generated by splitting C_i at M. Furthermore, if a B_i^m satisfies the size constraint(i.e., $|B_i^m| \leq L$) then it can be configured as a LA.

Corollary 2

If every gate node of a component C_i has the degree of one on C_i, the corresponding component cannot be an admissible union of more than one its subgraphs.

Property 1 is based on the fact that if two LAs, A^p and A^q are possibly configured on a given component C_i resulting from a backbone configuration, the common nodes of A^p and A^q should not only constitute a node-cut of C_i but also be gate nodes. Property 1 provides a way to configure LAs on a given component without increasing the size of the backbone. On the contrary, corollary 2 gives a condition when configuring more than one LA is impossible: if corollary 2 happens, the corresponding component can only be a LA as a whole.

3.3 Solution Methods

One general and practical approach to solving [LSHC] starts with an initial seed backbone profile, and grows the seed until a HC satisfying all the conditions (1) to (6) is found. Even though such a sequential update of the backbone profile does not guarantee an optimal solution, this heuristic approach has strong practical implications. First, since [LSHC] is only a sub-module of an entire package for scalable and reliable HC of a link-state AS([10]), several good solutions present often more value than one optimal solution. Second, given the role of the backbone in the link-state protocol, required is network administrators' preference for HC in the backbone selection process. Thus, incorporating flexibility in hierarchy design has practical importance, whereby heuristic approach gains a strong advantage over exact algorithm.

The heuristic procedure consists of the following major functions:

- Initialization: to get an initial seed of the backbone profile.
- Backbone update: to feed the backbone profile until all the resulting components become feasible.
- Feasibility condition check: to check the termination criteria of the heuristic by investigating feasibility of each component resulting from a temporary backbone profile.

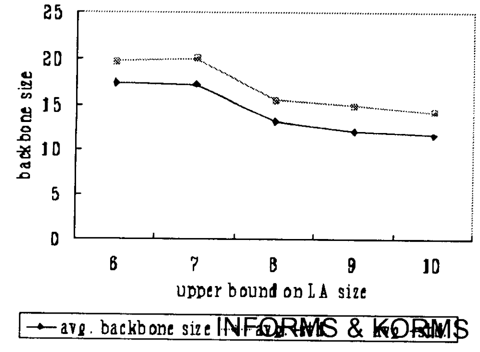
Even though property 1 can be used to identify all the possible LA configurations for a given component, the counting complexity to enumerate all the node-cuts composed of gate nodes only may hinder the exact feasibility check algorithm from practical implementation. Therefore, we employ in this paper a heuristic procedure that makes the feasibility check simple. That is, after identifying every articulation gate node of a component through DFS(Depth First Search), we form an admissible union of blocks $\{B_i^1, \ldots, B_i^m\}$ and check the size condition for each B_i^k, $k = 1, \ldots, m$.

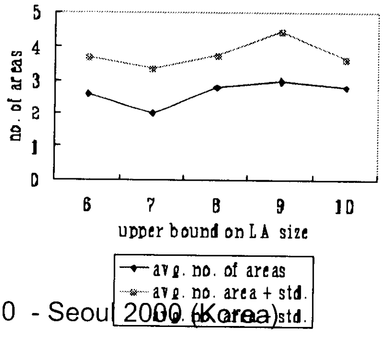
3.3.1 Example and Simulation Results

Table 1 summarizes experimental results on the same network by changing initialization and parameter L, the upper bound on the LA size. For every 10 repetitions of the experiment with L fixed at the same level, the heuristic finds at least one optimal solution to [LSHC]. In the light of the standard deviations of the results, the initial choice of a backbone profile affects the overall performance of the heuristic. We can see in figure 1 that as L increases, average backbone size decreases, but average number of areas seems rather independent of L.

Table 1: Experiment results

Upper bound on LA size (L)	6	7	8	9	10
Average backbone size (standard deviation)	17.3 (2.41)	17.2 (2.74)	13.1 (2.47)	12 (2.91)	11.5 (2.72)
Average number of LAs (standard deviation)	2.6 (1.07)	2 (1.33)	2.8 (0.92)	3 (1.41)	2.8 (0.79)
The largest size of LAs(average)	5.6	6.1	7.5	8.9	9.4





(a) result 1: backbone size (b) result 2: number of areas

Figure 1: Experimental results

4. Concluding Remarks

Critical design principles for successful hierarchical implementation of a link-state routing protocol have been addressed together with a model and heuristics to formulate and solve the issues. The first and most important decision when designing a HC on the link-state network is to determine the boundary of the backbone, thereafter feasible LA structures are configured. Ensuring that these activities are properly planned and executed will make all the differences in link-state implementation since the network performance in terms of scalability and stability depends on HC.

The proposed framework and solution method provide more general rules and flexibility in introducing hierarchy. For more efficient and situation-specific implementation of [LSHC], we have to extend the model and solution method. For instance, a network administrator may enforce a set of routing policies based on geographical or organizational grounds on his/her network; also, for reliable access from LA, LA should have multiple gate routers to prevent disconnection due to failure in primary gate router; we can also incorporate special restrictions on the backbone profile(such as restriction on the backbone topology) with a little modification of the suggested solution methods.

REFERENCES

1. Moy, J.T. (1998) OSPF: anatomy of an Internet Routing Protocol, Addison Wesley
2. Thomas, M.T. II. (1998), OSPF Network Design Solutions, Cisco Press
3. Jaffe, J.M. (1988) Hierarchical clustering with topology database, Computer Networks and ISDN Systems, 15, 329–339
4. Tsai, W.T., Ramamoorthy, C.V., Tsai, W.K., and Nishiguchi, O. (1989) An adaptive hierarchical routing protocol, IEEE Transaction on Computers, 38 1059–1075
5. Moy, J. (1991) OSPF protocol analysis, Request For Comments #1245
6. Moy, J. (1995) OSPF database overflow, Request For Comments #1765
7. Behrens, J. and Garcia-Luna-Aceves, J.J. (1998) Hierarchical routing using link vectors, Proceedings of IEEE INFOCOM 98, 702–710
8. Moy, J.T. (1994) Multicast routing extension for OSPF, Communications of the ACM, 37, 61–67
9. Halabi, S. (1996) OSPF design guide, Cisco Systems Network Supported Accounts(white paper)
10. Kim, D. and Tcha, D.W., Scalable domain partitioning in Internet OSPF routing, Telecommunications System, forth-coming

11. Zegura, E.W., Calvert, K.L., and Bhattacharjee, S. (1996) How to model an internetwork, Proceedings of IEEE INFOCOM 96, 594–602