IP Address Consolidation in Enterprise Networks

Sarvesh Chandra (190773)
Electrical Engineering
IIT Kanpur
Kanpur, India
csarvesh@iitk.ac.in

Amitangshu Pal
Assistant Professor
Computer Science and Engineering
IIT Kanpur
Kanpur, India
amitangshu@cse.iitk.ac.in

***Abstract*—It is a common industry practice to use Private IP addresses in an enterprise network. Different locations of the same enterprise may use the same IP address ranges if they use distinct networks. Different enterprises often use the same IP ranges within their own enterprise networks. During the merging of different locations within the same enterprise or different enterprises, conflicts may be detected making routing inefficient thereby creating a need for them to be resolved. Furthermore, consolidation of the subnets needs to be done to minimize the size of the resultant routing tables, thereby making routing efficient. The two problems are NP-hard and the proposed method employs specific algorithms and heuristics to solve them. Following the proposed method, the number of routing table entries is shown to get minimized by 80-90% by changing only 3-4% of the subnet addresses.**

I. Introduction

IP addresses in enterprise networks are often not assigned systematically, without any focus on minimizing routing table sizes. The problem becomes particularly prominent in large companies having multiple global locations, where multiple locations may even have overlapping subnets causing routing problems. These problems usually come to notice only when a user experiences them. Manual correction to these problems only increase the complexity of the problem because reassignment may lead to more conflicts.

The problem further amplifies when two companies (or part thereof) merge. Overlapping subnets becomes a significant problem here and leads to endless looping and routing fallacies. Even the size of the routing tables increases to a great extent during mergers, which adds to routing inefficiencies. A partial merger exacerbates the problem because additional routing policies have to be determined so as to ensure that no route from the merging part reaching the merging firm goes through the part not merging (via firewalls). Existing corporate solutions are unstructured and create endless routing problems.

In this paper, we implement a solution to the problems of conflicts and increasing routing table sizes via suitable algorithms. We will attempt to implement a method to resolve all conflicts arising out of mergers as well as consolidate the address space of two merging enterprises to minimize routing table sizes.

II. Conflict Resolution

The first step in the resolution of the problems arising out of mergers is the making of a conflict-free address space in the enterprise network. After merger, we need to find the subnet pairs which are conflicting across different locations. The enterprise network used for our implementation uses BGP (Border Gateway Protocol), thus each location in the network has a corresponding AS (Autonomous System) number. We thus figure out the conflicting subnets between each pair of distinct AS numbers in the network. To resolve the conflicts, the address spaces of these subnets need to be re-assigned within the network while also minimizing the reconfiguration of subnets which are more critical to the network. For instance, changing DHCP addresses will have the least relative importance compared to network and critical IPs because their re-assignment would only require the administrator to change the corresponding values in the DHCP server.

We represent each subnet as a vertex of a graph G = (V, E), there are edges in between the vertices if they conflict. We call G the conflict graph where V and E are the set of vertices and edges respectively. The cost (weight) of a subnet (vertex) depends on two factors, the number of physical entities that needs to be manually changed and their relative importance depending on their types. To retain the best possible combination of subnets (with maximum weights) we need to retain the maximum weight independent set (WIS) of graph G while we move the other subnets to other address space. As WIS is an NP-complete problem, we first formulate the integer linear program formulation of this problem and then relax the integrality constraint to propose a heuristic solution.

The weighted independent set (WIS) problem can be formulated in the integer linear programming as follows:

$$\begin{aligned} \max_{x} \quad & \sum_{i \epsilon V} w_i x_i \\ \text{s.t.} \quad & x_i + x_j \leq 1 \quad (i,j) \epsilon E \\ & x_i \epsilon \{0, 1\} \end{aligned} \tag{1}$$

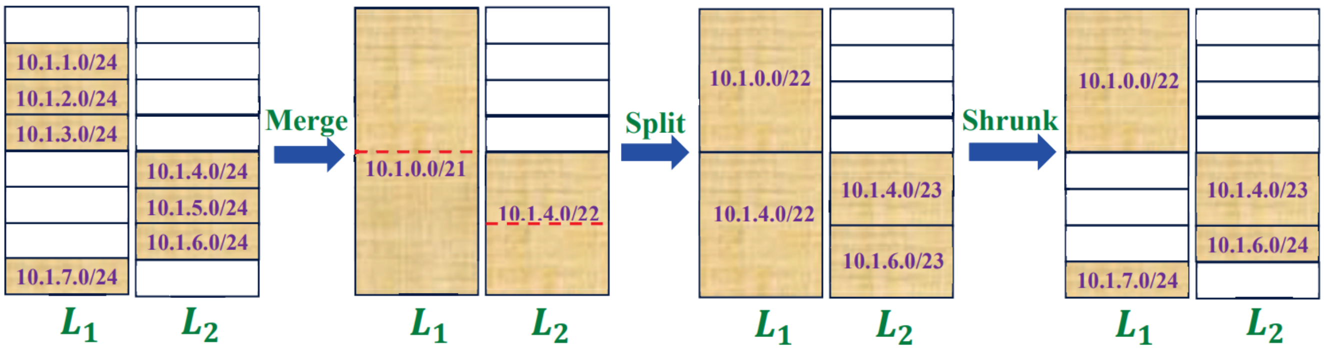

Fig. 1: An illustration of the manner in which merge and split proceeds

In the above problem, $w_i = \sum_t \eta_t^i w_t$ is the weight of vertex i and x_i is a binary decision variable which is 1 if vertex i is chosen and 0 otherwise. Also, η_t^i is the number of entities of type t that need to be touched in case subnet i is changed, and w_t is the relative importance of type t hosts.

The problem is NP-hard and approximation algorithms use the LP relaxation as follows:

$$\begin{aligned} \max_{x} \quad & \sum_{i \epsilon V} w_i x_i \\ \text{s.t.} \quad & x_i + x_j \leq 1 \quad (i,j) \epsilon E \\ & 0 \leq x_i \leq 1 \end{aligned} \tag{2}$$

This problem is related to the more common Weighted Vertex Cover problem [1] and its solution is such that we can divide the vertices into three sets $S_0, \quad S_{0.5}, \quad S_1$ with x_i equal to 0, 1/2 and 1 respectively.

$S_{0.5}$ will have no element in common with S_1. S_1 will be an independent set. We now form a subgraph consisting only of $S_{0.5}$ and the edges representing the conflicts between the elements of $S_{0.5}$. The weights are now:

$$d_v = \frac{\sum_{i \epsilon N_v} w_i}{w_v} \tag{3}$$

where N_v denotes neighbours of v in subgraph. The vertex with the minimum weighted degree is added to S_v and it, along with its neighbours, is removed from the subgraph. This is done until the subgraph is empty. The addition of minimum weighted degree vertex implies that the vertex has relative importance much greater than its conflicting neighbours and hence needs to be kept.

A. Reallocation of the Remaining Subnets

As the algorithm is removing some vertices from the original graph as well as the subgraph, these vertices need to be reallocated in the address space. The suggested algorithm for this is sorting of the subnets according to their masks and reallocating the remaining subnets in the vacant regions of the address space. The reallocation must be done in a way such that the effect of consolidation (the next step) on the size of the routing tables is maximized. This can be done by re-allocating the subnet only within the vacant regions occurring within the start and end IP addresses in a location if the size of those regions permit.

III. Address Space Consolidation

After conflict resolution, we will have an address space which will be conflict-free across different locations. We now move to the second stage where we can minimize the routing table sizes at each location by a suitable consolidation of the address space. We follow an iterative merge-split process. We sort the subnets within each location in the order in which they appear in the address space. We now begin the merge process by merging the last two subnets. The merge is acceptable if there is an increase in the utility function given as:

$$\phi(CS) = B(CS) - L(CS) = \alpha(M - |CS|) - \beta W(CS) \tag{4}$$

In simple terms, CS is the collection of the subnets in each location. B(CS) or the benefit function is proportional to the consolidation (reduction in number of subnets) while L(CS) or the loss function measures the amount of fresh conflicts occuring due to the merger.

$L(CS) = \beta W(CS)$ measures the amount of fresh conflicts occuring due to the merger by calculating the number of subnets that need to be changed (W(CS)) which is obtained as a result of applying the maximum weight independent set algorithm on the freshly obtained graph after the last merge or split.

In [Fig. 1], note that merging can happen only if the separation between the merging subnets is bounded by τ. If

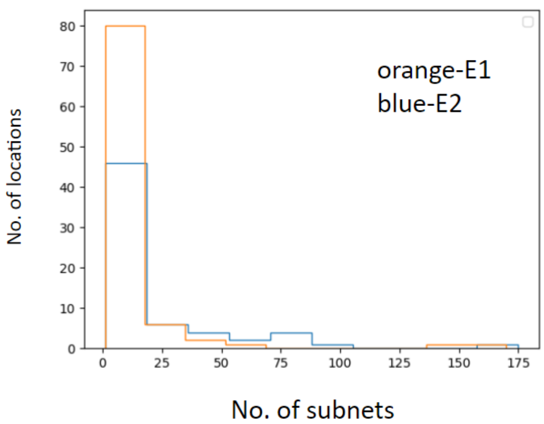

Fig. 2: Plot showing the no. of locations for every subnet size varying from 0 to 175

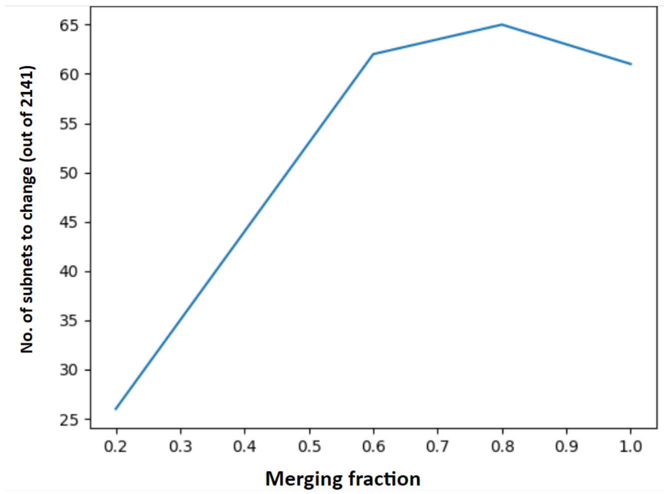

Fig. 3: Plot showing the no. of subnets needed to be changed against the merging fraction

the merge increases the utility, we replace the two merging subnets with the merged supernet in the address space of that location. If the merge fails, we proceed with the merger of the second and third last element, until the first subnet is reached.

After the merge process has ended in the first iteration, we begin the split process. Again starting with the last subnet, we split the subnets into two equally sized subnets if the utility function increases by doing so. If not, we continue to iterate on each of the subnets till the first subnet is reached. This merge and split completes the first iteration of the process. We continue this process for some iterations.

It must be noted that after all the iterations are over, we may be left with some fresh conflicts arising out of the newly created supernets. We employ the Maximum Weight Independent set algorithm again on the subnets to get a conflict free address space in the enterprise network.

IV. Experiments

We performed both conflict resolution and address consolidation by hypothetically merging the enterprise networks of two companies E1 and E2. The maximum number of subnets in any location was found to be 175 of E1 or E2. We plotted the distribution of subnets according to their sizes as in [2]. We observe from the histogram [Fig. 2] that there are only a few highly crowded locations.

Both E1 and E2 had some overlaps within themselves which were cleared. We introduced three types of subnets in every location while assigning a weight to each subnet. If the subnet mask was x, we chose the number of DHCP, network, critical hosts randomly between 0 and 0.50x, 0.20x, 0.05x respectively which ensured that the number of active hosts at any time is less than 70% of the subnet size, typical of enterprise networks. The weights assigned to them were 1, 2 and 10 respectively.

At the last step of address consolidation, we see that some of the subnets need to be changed for a conflict-free merger of E1 and E2. We plot the number of subnets needed to be changed against the merging fraction. A merging fraction of 0.2 denotes that the largest 20% subnets of E1 and E2 are being merged.

From this plot, it is noticed that the number of subnets needed to be changed does not change much after the merging fraction is made 0.6. This is justified on the basis of the skewed nature of the histogram of the number of subnets in the different locations, which proved that only very few subnets were highly crowded. We saw in [Fig. 2] that most of the locations have ¡40 subnets while very few locations have ¿50 subnets, with maximum number of subnets at any location being 175.

Across multiple trials of the suggested merger with random weights given to each subnet, we saw that the number of subnets that need to bechanged rises to a maximum of 85. Thus, the number of subnets that need to be changed seems to be 3-4% of the total number of subnets, which is fairly less.

We see in [Fig. 4] that for all values of the merging fraction, the resultant address space is consolidated by more than 60% of the total number of subnets in the merging locations and is close to 85% for a complete merger. This 85% consolidation is achieved with a cost of changing only 3-4% of the subnets, which is fair enough compared to the drastic reduction in routing table entries.

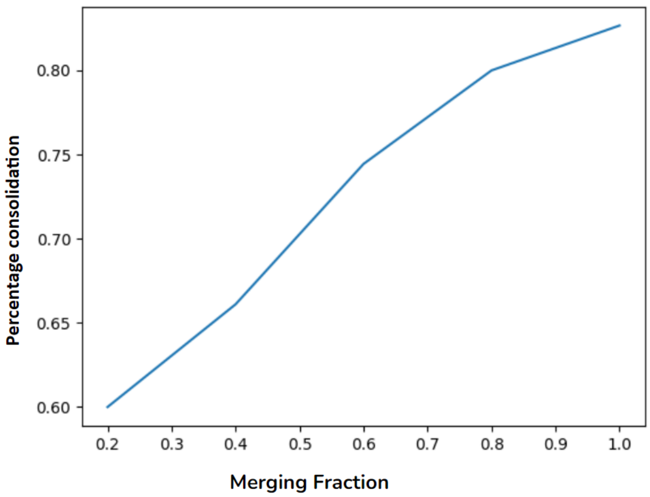

Fig. 4: Plot showing the no. of subnets needed to be changed against the merging fraction

V. Conclusion and Further Work

We devised algorithms for both conflict resolution and address consolidation. From the experiments we have have concluded that barely a change in 3-4% of the subnets can help consolidate the address space in an enterprise network by 80-90% and make it conflict-free.

This work leaves a scope of answering the question of a tradeoff between the extent of IP address changes and the consolidation. If the network administrator can substantially resolve the conflicts by changing only a few large subnets then that method is more suited to the problem than a minimization of IP address changes suggested here. We plan to study this issue as well as the anomalies that may occur during the process of reconfiguration (packet losses, etc.) and minimize their undesirable impact.

References

[1] T. H. A. Kako, T. Ono and M. M. Halldorsson, "Approximation algorithms for the weighted independent set problem," in *Graph-Theoretic Concepts in Computer Science, 31st International Workshop, WG 2005, Metz, France, June 23-25, 2005.* Revised Selected Papers, 2005, p. 341–350.

[2] A. P. I. El-Shekeil and K. Kant, "Ip address consolidation and reconfiguration in enterprise networks," *25th International Conference on Computer Communication and Networks (ICCCN), Waikoloa, HI*, pp. 1–9, 2016.