IP Address Consolidation and Reconfiguration In Enterprise Networks

(Invited Paper)

Ibrahim El-Shekeil, Amitangshu Pal, and Krishna Kant
Computer and Information Sciences, Temple University, Philadelphia, PA 19122
E-mail:{ielshekeil,amitangshu.pal,kkant}@temple.edu

***Abstract*—Private IP addressing is commonly used in enterprise networks. Different enterprises or even different locations/business units of the same enterprise may use the same IP address ranges as long while those networks are separate. Consequently, during mergers and acquisitions, or network consolidations within an enterprise, overlapped and conflicted IP segments (subnets) arise frequently. These must be identified and resolved to allow communication between any pair of source and destination hosts in the merged networks. Furthermore, the combined network may unnecessarily use many disparate IP address ranges which increases the size of the routing tables and makes routing integrity verification difficult. In this paper, we identify different conflict scenarios and consider ways of resolving those conflicts to minimize manual changes and to minimize the routing table sizes. The problem turns out to be NP-hard and rather complex, and we devise effective heuristics to solve the problem. By taking some real-world examples, we show that by changing ∼6-8% of the subnet addresses the outlined method can effectively resolve the subnet conflicts. The scheme also reduces the number of subnet entries by ∼80-90% by consolidating the subnet entries, which significantly reduces the routing table sizes.**

I. Introduction

The assignment of Internet Protocol (IP) addresses to various machines in a typical enterprise or data center is anything but systematic. In most cases, the IT systems grow organically with IP addresses assigned or reassigned haphazardly, based largely on what the administrators deem as appropriate or convenient. The result is a jumble of subnets with varying degrees of live addresses, an unnecessarily large number of routing table entries, and poor visibility into how the available address space within the organization is used. The problem becomes particularly severe with large enterprises that have multiple locations spread throughout the country or even the world. In this case, various locations may even use overlapping subnets which could cause routing anomalies. Often such anomalies are not detected until there is a real user complaint or IP address conflict. Unfortunately, even when observable problems expose the problems, no comprehensive action is taken; instead, the issue is resolved by making some quick and dirty IP address or routing table fixes, which in many cases further exacerbates the problem.

Mergers and acquisitions among companies of all sizes are routine in the business world, but the complexity of consolidating operations is increasingly dominated by the difficulty of aligning the digital assets at all levels including networks, firewalls, security policies, storage infrastructure, middleware and of course the applications. In particular, merging the enterprise networks creates challenges even if each individual network itself is free of the issues addressed above. The merging networks often have overlapping IP addresses spaces which would result in misrouting or endless looping of traffic if not corrected. Furthermore, many endpoint addresses may conflict and must be changed. Resolving these issues are further complicated in case of partial mergers. For example, company A may split itself into parts A1 and A2 with the express purpose of A2 merging with company B. In this case, the routes between A1 and A2 must be severed or restricted via suitable firewalls on each end. At the same time, it is necessary to establish new routes between A2 and B that do not go through A1. The existing misconfigurations, overlaps and conflicts within and across entities make this quite challenging. Often these changes too are done in an ad hoc and unstructured manner, which leads to extremely long transition periods (at least months, sometimes even years!) during which the clients may experience unreachability of certain destinations, large routing delays due to large number of routes, and unauthorized traffic flows leading to information leakage or vehicles for network attacks.

In this paper we address the issue of resolving conflicts, overlaps and consolidating the subnets in large organizations such that the reassignment of IP addresses (which often must be done manually) is minimized and so is the number of distinct routes that must be recorded in the routing table. It is easily shown that these problems are NP-hard; therefore, it is necessary to develop effective heuristics that is scalable. Although the initial motivation for our work resulted from mergers and acquisitions, the real world data shows that these problems are prevalent even within single businesses, particularly those with multiple geographically distributed sites. Furthermore, issues similar to merger/acquisition arise as business units or locations of a single company are reorganized. It follows that the solutions we propose could be used much more frequently than the merger/acquisition scenario would imply.

This research was supported by the NSF grant CCF-1407876.

978-1-5090-2279-3/16/$31.00 ©2016 IEEE

In view of the above, it is important to consider not only the quality of the solution but also the reconfiguration time during which some traffic may be mishandled. It is important to note that a proper IP address is crucial not only for routers and end-points, but also for a long list of other critical resources. These include DHCP servers, authentication servers, firewalls, NAT devices, load balancers, DNS servers, IP-sec/VPN appliances, proxy servers, reverse proxy servers, iSCSI HBA or target devices, accelerators, various types of management interfaces (e.g., for switches, storage controllers, or appliances), etc. The mishandling of traffic by many of these devices can be minimized via a proper ordering of activities; for example, if we install a new entry in the firewall corresponding to the new IP address before actually changing the IP address, we can prevent the firewall from discarding the traffic to the new IP target. However, the issue of configuration change ordering is beyond the scope of this paper and is addressed in [1].

In this paper our main contribution is as follows: we address the resolution of IP address overlaps/conflicts and consolidation of the address space, which are two interrelated operations and must be handled together to yield minimal change and best compaction. As we shall show, each of these is an NP-hard problem in itself; therefore, a combined solution is extremely complex. Thus, we address the problem in two stages – conflict resolution followed by consolidation. Such a technique is more practical since the IP address change (needed for conflict removal) often requires manual intervention, and is best done entirely before moving on to the consolidation part that can be run largely in an automated fashion. Our simulation results show the proposed scheme removes the address conflicts by just changing $\sim$6-8% of the subnet addresses in case of merging of two enterprises from real traces. On the other hand, the outlined address space consolidation scheme drastically reduces the number of subnet entries by $\sim$80-90%. To the best of our knowledge, this is the first work dealing with the issues of enterprise merging – one that tries to achieve address conflict removal and consolation in large enterprise environment with minimal manual overhead or intervention.

Accordingly, the outline of the paper is as follows. Section II discusses the conflict resolution and Section III discusses the IP address state consolidation issues including their complexity and heuristic solutions. Section IV reports results based on real routing table data obtained from two scenarios of actual merger between large companies. Section V discussed the related works. Finally, section VI concludes the discussion and discusses potential future work.

II. Address Conflict Resolution in Enterprise Merging

At the time of enterprise merger/acquisition or interconnection of disparate networks within one enterprise, several address conflicts may be detected. Typically, these are handled by filtering out the conflicted subnets from the routing tables and creating suitable entries in the Network Address Translation (NAT) appliance to hide the conflicted subnets. However the NATting solution by itself is cumbersome because of the need to maintain the filter for the routing policies and maintain NAT policies, as well as removing them after resolving the conflicts. Thus the long-term solution is to change some of the subnet addresses to another address space.

Subnet no	Location L	Subnet Addr	#Active IPs of type		
			1	2	3
s_1	1	10.1.0.0/24	10	12	0
s_2	1	10.1.1.0/24	200	1	1
s_3	1	10.1.2.0/24	200	0	0
s_4	1	10.1.3.0/24	200	0	0
s_5	2	10.2.0.0/24	200	0	0
s_6	2	10.1.3.0/24	200	10	0
s_7	2	10.2.2.0/24	200	10	0
s_8	2	10.2.3.0/24	200	0	0
s_9	3	10.1.0.0/22	500	0	0
s_{10}	3	10.2.1.0/24	0	10	0
s_{11}	4	10.1.3.0/24	0	0	100

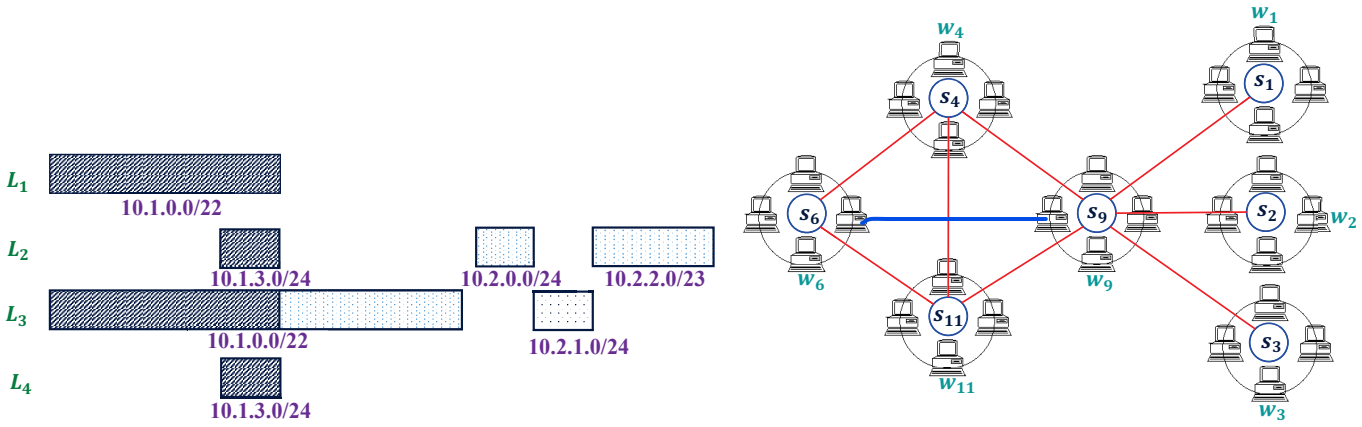
TABLE I
SUBNETS CONFLICTS

Conflicted IP subnets could be user subnets, network resource subnets like typical servers, printers etc., or critical resource subnets like DNS, NAT, authentication servers, etc. User subnets usually are dynamically assigned using a Dynamic Host Configuration Protocol (DHCP), whereas network or critical resource subnets usually are statically assigned and configured manually at each host. Thus resolving the conflicts among the subnets need to be decided by considering the number of assigned (active) IP addresses as well as their types. For example changing the static addresses needs manual intervention which is costly and time consuming, thus changing those is less preferable compared to that of DHCP subnets.

A. Identifying Subnets for Movement

While resolving the conflicts we need to change the addresses that incurs minimum cost. Fig. I shows an instance of the problem. Assume that after merging there are four locations $\{L_1, L_2, L_3, L_4\}$: each location L_i is a building or group of buildings that are connected in a Local Area Network (LAN). The locations have the following subnets: $L_1 = \{s_1, s_2, s_3, s_4\}$, $L_2 = \{s_5, s_6, s_7, s_8\}$, $L_3 = \{s_9, s_{10}\}$, and $L_4 = \{s_{11}\}$. We notice that subnets in location L_3 overlap with locations L_1, L_2 and L_4; also there are overlaps in between locations L_1, L_2 and L_4 itself. The conflicts in between the locations and the subnets are highlighted in Fig. 1(a) and Fig. 1(b) respectively.

To resolve the conflicts we need to change certain addresses such that the cost incurred due to this change is minimized. In Fig. 1(b) 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



(a) (b)

Fig. 1. (a) Subnet conflicts in Fig. I. (b) Construction of the conflict problem.

set of vertices and edges respectively. The cost (weight) of a subnet (vertex) depends on two factors:

(a) the number of physical entities that needs to be manually changed, and
(b) their relative importance depending on their types.

We express the weight of vertex i as $w_i = \sum_t \eta_i^t \omega^t$, where η_i^t is the number of entities of type t that need to be touched in case subnet i is changed, and ω^t is the relative importance of type t hosts. In case of DHCP, for making any changes in subnet address space, the administrator needs to change just in DHCP server which makes η_i^t equal to 1. In case of network and critical resources all the active hosts need to be configured manually, thus the η_i^t corresponding to these resources are equal to the number of active addresses. ω^t is highest for the critical resources and lowest for the DHCP addresses. This is because if the critical resources like DHCP, DNS, firewall etc. are down, the critical services cannot run. Thus changing the subnets of the critical resources costs the highest, which makes the ω^t corresponding to these resources higher.

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 [2], 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:

$$\begin{aligned} \text{Maximize} \quad & \sum_{i \in V} w_i x_i \\ \text{subject to} \quad & x_i + x_j \leq 1, \quad \forall (i,j) \in E \\ & x_i \in \{0,1\} \quad \forall i \in V \end{aligned} \tag{1}$$

where w_i 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. We next use a two-stage heuristic solution that is proposed in [3]. In the first stage we solve the linear programming (LP) as follows:

$$\begin{aligned} \text{Maximize} \quad & \sum_{i \in V} w_i x_i \\ \text{subject to} \quad & x_i + x_j \leq 1, \quad \forall (i,j) \in E \\ & 0 \leq x_i \leq 1 \quad \forall i \in V \end{aligned} \tag{2}$$

The above linear program gives the optimal solution corresponding to the LP version of the WIS with each $x_i \in \{0, \frac{1}{2}, 1\}$ [3], [4]. We divide the vertices into three sets $S_0, S_{\frac{1}{2}}, S_1$ with x_i equal to 0, $\frac{1}{2}$ and 1 respectively. The vertices in S_1 are all independent (i.e. they do not have any common edge) and do not share any common edge with $S_{\frac{1}{2}}$. We first keep the vertices in S_1 in the independent set I.

We next take the induced subgraph H corresponding to the vertices in $S_{\frac{1}{2}}$ and their corresponding edges in between them and then apply the following scheme. Assume that N_v is the neighbor set of vertex v in H, then we define the *weighted degree* of vertex v as $d_v = \frac{\sum_{i \in N_v} w_i}{w_v}$. We next select the vertex in H with minimum weighted degree (ties are broken randomly) and include in I, delete the vertex and its neighbors from H. That is, if the weight w_v of vertex v is higher than the sum of weights of its neighbors $\sum_{i \in N_v} w_i$, then the vertex is included in I. Note that it is most fruitful to change the IP address of a vertex with the highest weighted degree, because a single such change would resolve many conflicts. The desired addition to I is then a node with smallest weighted degree. We repeat this process until the remaining subgraph becomes empty.

Lemma 1. *The approximation ratio of the above WIS heuristics scheme is* $\left(\frac{\bar{d}+1}{2}\right)$*, where* $\bar{d}$ *is the weighted average degree*

of G and is given by $\bar{d} = \frac{\sum_{v \in V} \sum_{i \in N_v} w_i}{\text{total weight}}$.

Proof. Please refer to [3] for the proof. □

An illustrative example: We generate the WIS corresponding to the conflict graph in Fig. 1(b). We assume all the weights to be 1 for simplicity. We solve the LP problem corresponding to the conflict graph in AMPL solver [5]. The LP problem generates a value of $\{1, 1, 1, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2}\}$ corresponding to the variables $\{x_1, x_2, x_3, x_4, x_6, x_9, x_{11}\}$. Thus after the first stage $I = S_1 = \{s_1, s_2, s_3\}$. Notice that the vertices of S_1 are independent and do not share any edge with that of $S_{\frac{1}{2}} = \{s_4, s_6, s_{11}\}$.

We next construct H with $\{s_4, s_6, s_{11}\}$ and generate their weighted degrees, which come out to be equal to 2 for all three vertices. We thus choose any one of these three randomly and include it to I (say s_4). We then remove the neighbors of s_4 which makes the remaining subgraph empty. Thus the algorithm stops with $I = \{s_1, s_2, s_3, s_4\}$, while the remaining subnets $\{s_6, s_9, s_{11}\}$ in the conflict graph are decided to be changed to avoid conflict.

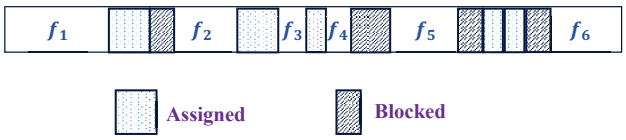


Fig. 2. Available address space at a location.

B. Reallocation of Subnets

We next need to reallocate the *remaining subnets* (that are not in I) to avoid conflicts. Keeping in mind that the address space will be summarized in the next stage, we try to reassign them in a manner that incurs maximum benefit. To do so we first sort the subnets in decreasing order of their sizes. Also assume that the fragments in the address space are denoted as $f_1, f_2, ..., f_n$. Fig. 2 shows the structure of the address space of a certain location. The *assigned* and *blocked* spaces are the address spaces that are already assigned subnets by the same and other locations respectively. The remaining subnets are assigned in order one after other using the following principle. For each fragment there will be atmost two possible positions where a subnet (say s_i) can be assigned (either at the left or right of the fragment as shown in Fig. 2). Thus for n fragments there will be atmost $2n$ possible positions, at each position the benefit of placing the subnet will be calculated. At any particular position, the benefit is calculated by summarizing the nearby subnets after positioning s_i and counting the reduction of the subnet entries after summarization. For example in Fig. I, moving s_6 from 10.1.3.0/24 to 10.2.1.0/24 makes the summary of 10.2.0.0/22 by summarizing $\{s_5, s_6, s_7, s_8\}$, thus the benefit of placing s_6 at 10.2.1.0/24 is $4 - 1 = 3$. The position which gives the highest benefit will be chosen to place the subnet. Also notice that while summarizing, we need to make sure that the new summary does not make any new conflicts. This process is repeated for all the remaining subnets.

III. Address Space Consolidation Stage

After the conflict resolution and reallocation stage the subnets are now in a non-conflicting stage. We thus start our address space consolidation stage to generate the summary addresses corresponding to different neighboring subnets, so that the routing table entries of the routers and gateways are largely minimized. We first generate a coalition formation game for this consolidation operation, an then resolve another level of conflicts that may arise due to the consolidation phase.

A. Preliminaries and Complexity of Subnet Consolidation

Definition 1. *A collection of coalitions in a grand coalition $\mathcal{N}$, denoted as C, is defined as the set $C = \{C_1, C_2, \ldots, C_l\}$ of mutually disjoint coalitions $C_i \subseteq \mathcal{N}$. In other words, a collection is a random group of disjoint coalitions C_i of $\mathcal{N}$ not necessarily including all players of $\mathcal{N}$. If the collection includes all player of $\mathcal{N}$, i.e. $\cup_{i=1}^{l} C_i = \mathcal{N}$, the collection is referred to as partition of $\mathcal{N}$.*

Definition 2. *A coalition structure over $\mathcal{N}$ is a collection of coalitions $CS = \{C_1, ..., C_k\}$ such that $\bigcup CS = \mathcal{N}$ and $C_i \cap C_j = \phi$ for any $i, j \in \{1, 2, ..., k\} : i \neq j$. The set of all coalition structures over $\mathcal{N}$ is denoted as $\Pi^{\mathcal{N}}$.*

The *coalition structure generation problem* [6] is to find a coalition structure over $\mathcal{N}$ that maximizes the social utility of the coalition structure, i.e. if CS^* is the optimal coalition structure and ϕ is the value function of a coalition structure then $CS^* = \arg\max_{CS \in \Pi^{\mathcal{N}}} \phi(CS)$.

Our *subnet consolidation* problem is identical to the coalition structure generation problem by assuming the initial set of subnets as agent set $\mathcal{N}$ and the merging of the subnets into multiple supernets is a coalition structure CS. Thus the problem boils down to find out the optimal set of supernet/coalition generation to maximize the social utility. As the optimal coalition structure generation problem is NP-hard, our subnet consolidation problem is NP-hard too. This is because of the fact that the number of possible coalition structures is given by the Bell number, which exponentially grows with $|\mathcal{N}|$ [6].

B. Formulation of Iterative Coalition Formation Game

We first sort the subnet addresses corresponding to each location so that the neighboring subnets can merge together to form a supernet. A social utility based coalition formation game is then designed using which the subnets can form coalitions or supernets. Let us define a coalition game

$\mathcal{G} = (\mathcal{N}, \phi)$, where $\mathcal{N}$ is the set of players representing each subnet and ϕ is the social welfare function described later on. However determining the optimal coalition formulation is NP-hard, as it requires iterating over all possible coalitions of $\mathcal{N}$ subnets. Thus, an iterative coalition formation approach is more desirable. In [7], the authors have provided a generic framework for forming coalition games among players using two simple merge and split rules, which can be applied in an iterative fashion. However, before the framework can be adapted and applied to the problem, the following preference operator needs to be defined.

Definition 3. *A preference operator or comparison relation $\triangleright$ is defined for comparing two collections $O = \{O_1, O_2, \ldots, O_r\}$ and $P = \{P_1, P_2, \ldots, P_s\}$ with the same set of players $\mathcal{N}$ (i.e. $\bigcup_{i=1}^{r} O_i = \bigcup_{j=1}^{s} P_j \subseteq \mathcal{N}$). Thus, $O \triangleright P$, implies that the way O partitions $\mathcal{N}$ is preferred over the way P partitions $\mathcal{N}$.*

The comparison relation ($\triangleright$) provides players with a way to compare coalitions before joining or splitting from them. Intuitively, this means that players in coalitions in P have the incentive to deviate and form coalitions given by O if doing so improves the social outcome, i.e.

$$O \triangleright P \Leftrightarrow \phi(O) > \phi(P) \tag{3}$$

Based on the above formulations, the rules for merging and splitting of the coalitions are defined as follows.

- Merge Rule: Merge the coalitions $\{C_1, \ldots, C_l\}$ if $\{\bigcup_{z=1}^{l} C_z\} \triangleright \{C_1, \ldots, C_l\}$
- Split Rule: Split the coalition $\{\bigcup_{z=1}^{l} C_z\}$ if $\{C_1, \ldots, C_l\} \triangleright \{\bigcup_{z=1}^{l} C_z\}$

The merge rule specifies that if the coalitions $\{C_1, \ldots, C_l\}$ can achieve a higher social welfare by merging, then they will merge and form a single coalition $\{\bigcup_{z=1}^{l} C_z\}$. Similarly, the split rule specifies that if the coalition $\{\bigcup_{z=1}^{l} C_z\}$ can achieve a higher social utility by splitting the coalitions into smaller coalitions, then they will split and form smaller coalitions $\{C_1, \ldots, C_l\}$.

C. Social Utility Function Generation

The proposed subnet consolidation problem can be modeled as a $(\mathcal{N}, \phi)$ coalition game where $\mathcal{N}$ is the set of players (the subnets) and ϕ is the utility function or value of a coalition. The value $\phi(CS)$ of a coalition structure $CS \in \Pi^{\mathcal{N}}$ must capture the trade off between the amount of subnet consolidation and the amount of overlap. For this purpose, $\phi(CS)$ must be an increasing function of the amount of subnet consolidation which can be represented as a benefit function of the number of subnets $B(CS)$ and a decreasing function of the amount of overlap which can be represented as a loss function $L(CS)$. A suitable utility function is given by

$$\phi(CS) = B(CS) - L(CS) = \alpha\left(\mathbb{M} - |CS|\right) - \beta.\mathcal{W}\left(CS\right) \tag{4}$$

where $\mathbb{M}$ is assumed to be the initial number of subnets (which we assume to be the maximum number of subnets possible) and α and β are the weights of the benefit and loss functions respectively. To calculate the $\mathcal{W}(CS)$ of a coalition structure CS, we construct a conflict graph where the coalitions are the vertices and there is an edge in between two coalitions if they overlap. The weights of the vertices (coalitions) are the sum of the weights of the individual subnets in the corresponding coalitions. We next run the WIS to find out the sum of the weights of the coalitions that need to be removed from the conflict graph to make it conflict free, which is basically the cost of making a coalition structure CS conflict-free.

D. Implementation of the Coalition Game

We now describe our merge-and-split based coalition formation scheme that runs iteratively. In each iteration the proposed algorithm starts from a partition $T = \{T_1, T_2, ..., T_l\}$ of $\mathcal{N}$. During the adaptive coalition formation phase any random coalition can start with the merge process. For implementation purposes, we assume that initially the individual subnets form separate coalitions. We next visit the coalitions sequentially, i.e. first $T_1 \in T$ starts the merge by attempting to collaborate with a nearby coalition (say T_2), if doing so will improve the social utility. If merging occurs, a new coalition $\tilde{T_1}$ is formed. Next coalition $\tilde{T_1}$ will attempt to merge with a nearby coalition (say T_3) that can improve the social utility. The search ends by a final merged coalition T_1^{final} composed of T_1 and one or several of coalitions in its vicinity. Notice that $T_1^{\text{final}} = T_1$ if no merge occurred. The algorithm is repeated for the remaining $T_i \in T$ until all the coalitions have made their merge decisions, resulting in a final partition F.

Fig. 3 shows an example of the merge operation where 7 subnets are merged to form 2 summaries or coalitions. Notice that for practical purposes we do not merge subnets if they are separated by some threshold Γ. This is because by merging two subnets that are largely separated will result in a significant number of unused spaces or fragmentation which need to be controlled by tunning the parameter Γ.

Following the merge process, the coalitions in the resulting partition F are next subject to split operations, if doing so will improve the social utility. The IP supernets can be split in different ways depending on the splitting position. For simplicity we assume that the split operation happens at the middle of the coalitions. After a split operation, a supernet is split into two coalitions, at each coalition the subnets are again shrunk to form a more compressed subnet structure. The smaller shrunk coalitions are again split if doing so will improve the utility. This is depicted in Fig. 3 where the coalitions 10.1.0.0/21 is

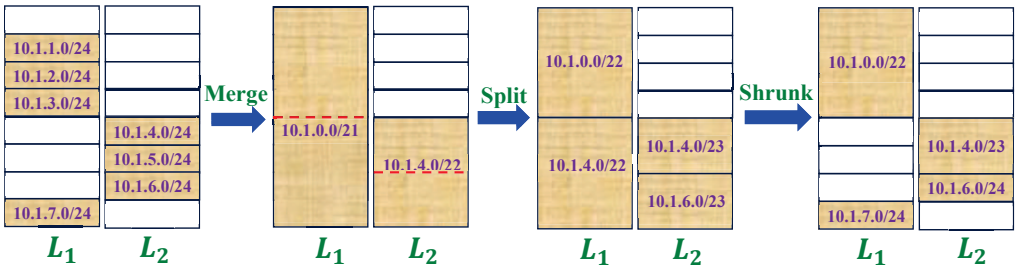


Fig. 3. Example of merge and split operation in the coalition game. The splitting points are shown in red lines.

first split from the middle into 10.1.0.0/22 and 10.1.4.0/22. The second coalition is then shrunk to form a more compressed coalition that consists of just one subnet 10.1.7.0/24. Thus an iteration consisting of multiple successive merge-and-split operations which is repeated until no improvement is noticed or the scheme terminates.

Theorem 2. *In the game $\mathcal{G}$, random iterations of the merge and split rule will always terminate in finite steps from any arbitrary starting point.*

Proof. Notice that at every iterations the merge and split rules generates a partition that improves the social utility, i.e. if the sequence of partitions are $P_1, P_2, ...$ then $P_{i+1} \triangleright P_i$, $\forall i$. However the number of different partitions are finite, thus the random iterations of merge and split rule will always terminate from any arbitrary point in finite steps. □

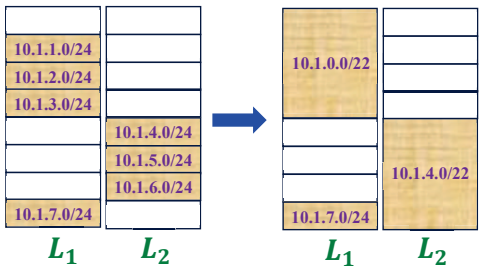


Fig. 4. Example of a new conflict after merge-split based coalition formation stage.

E. Resolving Conflicting IP Summaries

After the coalition formation stage there may be some overlapping summary addresses among different locations which needs to be reassigned. For example in Fig. 4 the merge-split coalition formation may result in summary addresses of 10.1.4.0/22 and 10.1.7.0/24 which overlap mutually. In such scenario we need to move either one of these two summary addresses in a separate address space. This problem is similar to that in section II but differs in one aspect: in section II the individual subnets were moved to avoid conflict, whereas in this section we will move an entire coalition to avoid conflict.

Scenario	Locations	Subnets
E_1	66	2473
E_2	113	1726
$E_1 + E_2$	179	4199

TABLE II
SOME STATISTICS FOR A REAL-WORLD MERGER

The intuition behind this is that before the coalition formation stage the subnets were absolutely non-conflicting. However after the coalition formation stage few subnets decided to merge together and form a coalition for a better utility, thus those subnets have sufficient incentives to stay together and so are required to move together.

We thus construct a WIS graph where the coalitions are the vertices rather than the subnets, and the weights are the cumulative weights of all the subnets in that coalitions. We thus use the same heuristics to solve the WIS problem as discussed in section II. After the WIS stage the coalitions that are required to be moved are reassigned in any vacant address space to avoid conflict.

IV. EXPERIMENTAL RESULTS

We evaluate the performance of our proposed schemes using simulations that exploit real-world data obtained in connection with two recent mergers, each of which involved two fairly large companies. In one case, only a part of one company merged with another thereby creating a scenario discussed in section I. Unfortunately, we are not at liberty to identify the companies involved in the two mergers. In the following we discuss results using data for one of the mergers involving companies that we will denote simply as E_1 and E_2. We started with the routing table entries of these before the merger and first removed all public addresses. It was noticed that E_2 had a few overlaps within itself; we removed the larger subnets corresponding to these overlaps. Table II shows the statistics for E_1 and E_2 following the overlap removal.

The available data did not include a detailed list of endpoint IP addresses, which is often difficult to collect. Therefore, we resorted to generating IP addresses within each subnet as follows: if the size of a subnet is x (obtained from

the subnet mask), then the number of DHCP, network, and critical hosts are obtained by randomly selecting each address between 0 and $0.5x, 0.25x, 0.05x$ with a probability of 0.5. This ensures that the number of active hosts at any subnet is no more than 75% of its size, which is representative of typical enterprises. Here ω^t corresponding to DHCP, network and critical resources are assumed to be 1, 2 and 10 respectively.

We record (a) the percentage of subnets that need to be moved to make the enterprise network conflict-free after merging, and (b) the amount of subnet consolidation which is the percentage of entries that are reduced after the merge-split based coalition game. We also generate multiple synthetic traces from the original trace by varying the number of locations in each enterprise and their corresponding table entries.

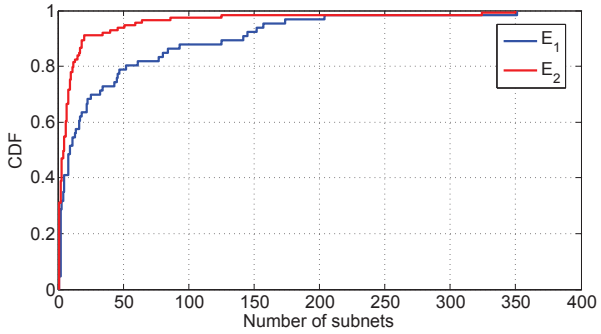


Fig. 5. CDF of the number of subnets at different locations.

Fig. 5 shows the distribution of the number of subnets in different locations at the enterprises E_1 and E_2. From this figure we can observe that the distribution is highly skewed; ∼80-90% of the locations have <50 subnets, whereas the maximum number of subnets at any location is around 350. Thus in the trace files few locations are highly crowded, whereas most of the locations have very small number of subnets.

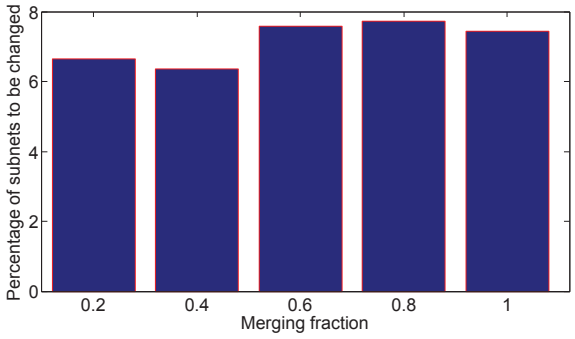


Fig. 6. Percentage of subnets that are to be changed to make the enterprise networks conflict free after E_1 and E_2 merges.

Fig. 6 shows the percentage of subnets that need to be changed for a conflict-free merging of E_1 and E_2 with the variation of the fraction of locations that are considered to be merged from these two enterprises. For example, the point 0.2 in the x-axis represents the case when the *largest* (w.r.t the number of subnets) 20% of the locations of E_1 and E_2 are assumed to be merged. Thus the point 1.00 in the x-axis represents the case when entire E_1 and E_2 merge; we call this number *merging fraction* for the sake of explanation. We assume the merger threshold Γ to be 512 for Fig. 6. From Fig. 6 we can observe that the number of subnets that needs to be changed varies from 6-8% when the merging fraction varies from 0.2 to 1. This shows that by changing few subnet addresses the enterprises can effectively remove all conflicts, which shows the effectiveness and the usefulness of the proposed scheme. Also notice that the amount of change needed does not vary significantly with the increase in merging fraction, which results from the highly skewed nature of the subnet distribution in these enterprises.

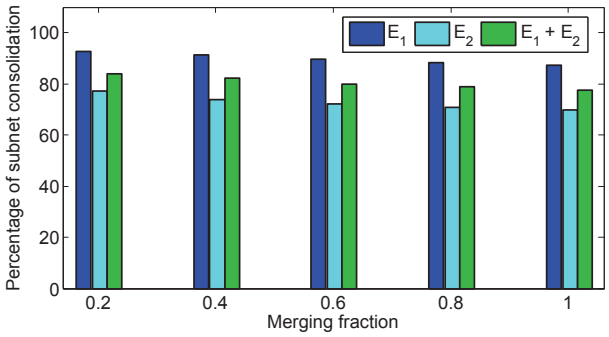


Fig. 7. Percentage of entries reduced after the merge-split based coalition game.

Fig. 7 shows the percentage of subnet entries that are reduced after the consolidation process with different merging fractions, while assuming Γ equal to 512. From Fig. 7 we can observe that through the iterative merge-split process we can reduce the subnet entries by ∼80-90% with different merging fraction scenarios. As we have mentioned that the subnet consolidation is also effective even within an enterprise, which is also reflected in Fig. 7. The amount of improvement does not change significantly with the increasing merging fraction due to its largely skewed nature.

V. Related Works

A. IP Conflict Resolution

In the IP address conflict resolution, much work has been done in the context of mobile ad-hoc networks (MANET) where merging and splitting of nodes from a MANET is common. However, we have not seen any algorithmic work on optimizing conflict resolution and subnet consolidation in enterprise networking context. The conflict resolution in MANET can be classified into two broad categories: centralized and decentralized. In the centralized scheme, a leader is elected as an address authority that maintains the state information of all the nodes in the MANET. Whenever a new

node joins the network, it needs to negotiate with the leader so that there will not be any address conflict. Some of the proposals that fall in this category are DACP [8], ODACP [9], VASM [10], Lightweight secure address configuration [11] etc.

In the distributed approaches a new node randomly chooses a tentative address and floods an address request message in the network to query for the usage of its tentative address. If the address is already in use, an address reply message is unicast back to the requesting node so that a different address can be selected. The absence of an address reply indicates the availability of the requested address. Such approaches are addressed in the proposals like AAA [12], MANETCONF [13], PRIME DHCP [14], AIPAC [15], MMIP [16], ADIP [17], IDDIP [18], IDSDDIP [19], secure auto-configuration [20], [21], quadratic residue based address allocation [22] etc. WeakDAD [23] is a slightly different procedure where every node is identified by a unique tuple consists of its IP address and a unique key which is typically its MAC address. While merging if two nodes have the same IP address, they can still be identified by their unique keys. Notice that our conflict resolution problem is significantly different than the above literature because in our problem the entire conflicting subnets with different weights need to be altered rather than individual nodes as done in MANET. Also in the above schemes the newly joined nodes generate fresh IP addresses that are not used by the existing MANET nodes, whereas in our scheme we try to minimize the overhead/cost of changing addresses in case of enterprise merging.

B. Coalition Games

There are two distinct classes of games that differ in terms of the factors that influence a coalition's value. The first class is called *Characteristic function games (CFGs)* where the value of a coalition depends solely on the identities of its members. The second class is called *Partition function games (PFGs)* where the value of a coalition depends on both the identities of its members as well as the way non-members are partitioned. In the literature on algorithmic game theory the CFGs have received more attention. In [24], [25], [26] the authors have adopted dynamic programming algorithm for the coalition structure generation problem, whereas the authors in [27], [28] have used integer partition-based representation for coalition structure generation. Several iterative, decentralized and greedy algorithms are proposed in [29], [30], [31], [32].

Metaheuristics like genetic algorithm and simulated annealing have been proposed in [33], [34]. Other greedy and adaptive searches are used in [35], [36]. Distributed Constraint Optimization framework has been adopted in [37], [38], [39] where the agents in the coalitions choose an action so as to maximize the sum of the reward functions from their choice of actions. In [7], [40], [41] the authors have described the notion of graph-restricted games where the nodes of the graph represent the agents, while the edges can be interpreted as communication channels, or trust relationships, which facilitate the cooperation. Authors in [42] have proposed an edge contraction based scheme by either removing an edge from a graph, or merging the two nodes that were previously joined by that edge.

In the networking research, CFG based coalition formation games have been studied in the context of cognitive radio networks [43], [44], small cell networks [45], vehicular networks [46], [47], file sharing networks [48], delay tolerant networks [47], mobile social networks [49] etc. On the other hand PFG games are comparatively less studied in the literature, however some of the proposals are found in [50], [51], [52]. Interestingly our subnet cosolidation problem falls under the category of PFG games as the utility of a coalition (or social/global utility in our case) is dependent on the subnets of the other coalitions due to the overlap of the subnets in between different coalitions.

VI. Conclusions and Future Works

In this paper we addressed the problem of resolving IP address conflicts and consolidating the IP address space in large enterprises, especially in the context of merger/acquisition of companies and the required alignment of their IT infrastructures. We proposed a conflict resolution scheme that moves a few subnets with the intent to minimize cost and overhead of the change. We also developed an address space consolidation mechanism so that the routing entries of the routers and gateways are largely minimized. The results show that the proposed scheme can substantially reduce the effort required in the conflict removal and result in much smaller routing tables.

This paper is a preliminary study to overcome the address space issues in large enterprises especially at the time of merging, without *aggressively* changing the subnet addresses. In the future, we plan to consider the broader question of tradeoff between the extent of IP address changes and the resulting reduction in number of subnets (i.e., number of routing table entries). For example, if the administrators are willing to change IP addresses of a few larger subnets in a particular way, it might allow much better compaction than the approach that simply minimizes IP address changes. We will also examine more complex situations that involve simultaneous merger and split of multiple companies.

In such situations, the order in which splits and mergers are handled can significant and an optimal ordering can reduce the perturbation substantially. Finally, we will examine the anomalies (e.g., unreachability, packet looping, packet loss) *during* the process of conflict resolution/consolidation, and how the undesirable impacts can be minimized. This would likely lead to on-line methods of conflict resolution/consolidation that can be run concurrently with the normal operation of the network for transacting business or serving customers. Finally, we plan to do a more thorough evaluation of our proposed method based on data from a much larger set of real-world scenarios.

REFERENCES

[1] I. El-Shekeil, A. Pal, and K. Kant, "Progressive recovery of interdependent services in enterprise networks," submitted for publication.

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1990.

[3] A. Kako, T. Ono, T. Hirata, and M. M. Halldórsson, "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, pp. 341–350.

[4] V. V. Vazirani, *Approximation Algorithms*. Springer-Verlag New York, Inc., 2001.

[5] AMPL, http://ampl.com/products/solvers/.

[6] T. Rahwan, T. P. Michalak, M. Wooldridge, and N. R. Jennings, "Coalition structure generation: A survey," *Artif. Intell.*, vol. 229, pp. 139–174, 2015.

[7] K. R. Apt and A. Witzel, "A generic approach to coalition formation," *International Game Theory Review*, vol. 11, no. 3, pp. 347–367, 2009.

[8] Y. Sun and E. M. Belding-royer, "Dynamic address configuration in mobile ad hoc networks," Tech. Rep., 2003.

[9] Y. Sun and E. M. Belding-Royer, "A study of dynamic addressing techniques in mobile ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 4, no. 3, pp. 315–329, 2004.

[10] M. Taghiloo, M. Dehghan, J. Taghiloo, and M. Fazio, "New approach for address auto-configuration in manet based on virtual address space mapping (vasm)," in *ICTTA*, 2008.

[11] M. Tajamolian, M. Taghiloo, and M. Tajamolian, "Lightweight secure IP address auto-configuration based on VASM," in *23rd International Conference on Advanced Information Networking and Applications, AINA 2009, Workshops Proceedings, Bradford, United Kingdom, May 26-29, 2009*, 2009, pp. 176–180.

[12] C. Perkins, J. Malinen, R. Wakikawa, and E. Belding-Royer, "Ad hoc address autoconfiguration," *IETF Internet Draft*, 2011.

[13] S. Nesargi and R. Prakash, "Manetconf: Configuration of hosts in a mobile ad hoc network," in *IEEE INFOCOM*, 2002.

[14] Y.-Y. Hsu and C.-C. Tseng, "Prime dhcp: a prime numbering address allocation mechanism for manets." *IEEE Communications Letters*, vol. 9, no. 8, pp. 712–714, 2005.

[15] M. Fazio, M. Villari, and A. Puliafito, "AIPAC: automatic IP address configuration in mobile ad hoc networks," *Computer Communications*, vol. 29, no. 8, pp. 1189–1200, 2006.

[16] U. Ghosh and R. Datta, "Mmip: a new dynamic ip configuration scheme with mac address mapping for mobile ad hoc networks," in *NCC*, 2009.

[17] ——, "ADIP: an improved authenticated dynamic IP configuration scheme for mobile ad hoc networks," *IJUWBCS*, vol. 1, no. 2, pp. 102–117, 2009.

[18] ——, "A secure dynamic ip configuration scheme for mobile ad hoc networks." *Ad Hoc Networks*, vol. 9, no. 7, pp. 1327–1342, 2011.

[19] ——.

[20] P. Wang, D. S. Reeves, and P. Ning, "Secure address auto-con.guration for mobile ad hoc networks," in *MobiQuitous 2005*, 2005, pp. 519–522.

[21] A. R. Cavalli and J. Orset, "Secure hosts autoconfiguration in mobile ad hoc networks," in *24th International Conference on Distributed Computing Systems Workshops (ICDCS 2004 Workshops), 23-24 March 2004, Hachioji, Tokyo, Japan*, 2004, pp. 809–814.

[22] X. Chu, Y. Sun, K. Xu, Z. Sakander, and J. Liu, "Quadratic residue based address allocation for mobile ad hoc networks," in *IEEE ICC*, 2008, pp. 2343–2347.

[23] N. H. Vaidya, "Weak duplicate address detection in mobile ad hoc networks," in *ACM MobiHoc*, 2002, pp. 206–216.

[24] D. Yun Yeh, "A dynamic programming approach to the complete set partitioning problem," *BIT Numerical Mathematics*, vol. 26, no. 4, pp. 467–474, 1986.

[25] T. Rahwan and N. R. Jennings, "An improved dynamic programming algorithm for coalition structure generation," in *Int. joint Conf. on Autonomous Agents and Multi-agent Systems (AAMAS)*, 2008, pp. 1417–1420.

[26] A. Björklund, T. Husfeldt, and M. Koivisto, "Set partitioning via inclusion-exclusion," *SIAM J. Comput.*, vol. 39, no. 2, pp. 546–563, 2009.

[27] T. Rahwan, S. D. Ramchurn, N. R. Jennings, and A. Giovannucci, "An anytime algorithm for optimal coalition structure generation," *J. Artif. Intell. Res. (JAIR)*, vol. 34, pp. 521–567, 2009.

[28] T. Michalak, J. Sroka, T. Rahwan, M. Wooldridge, P. McBurney, and N. R. Jennings, "A distributed algorithm for anytime coalition structure generation," in *International Conference on Autonomous Agents and Multiagent Systems*, 2010, pp. 1007–1014.

[29] O. Shehory and S. Kraus, "Coalition formation among autonomous agents: Strategies and complexity," in *MAAMAW*, 1993, pp. 56–72.

[30] O. Shehory, "Coalition formation methods in multi-agent environments," in *National Conference on Artificial Intelligence*, 1994.

[31] O. Shehory and S. Kraus, "Task allocation via coalition formation among autonomous agents," in *IJCAI*, 1995, pp. 655–661.

[32] M. Klusch and O. Shehory, "Coalition formation among rational information agents," in *European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, 1996, pp. 204–217.

[33] S. Sen and P. S. Dutta, "Searching for optimal coalition structures," in *International Conference on Multi-Agent Systems*, 2000, pp. 287–292.

[34] H. Keinnen, "Simulated annealing for multi-agent coalition formation." in *KES-AMSTA*, vol. 5559, 2009, pp. 30–39.

[35] N. Di Mauro, T. M. A. Basile, S. Ferilli, and F. Esposito, "Coalition structure generation with grasp," in *Artificial Intelligence: Methodology, Systems, and Applications*, 2010, pp. 111–120.

[36] A. Farinelli, M. Bicego, S. Ramchurn, and M. Zuchelli, "C-link: a hierarchical clustering approach to large-scale near-optimal coalition formation," 2013.

[37] P. J. Modi, "Distributed constraint optimization for multiagent systems," Ph.D. dissertation, Los Angeles, CA, USA, 2003.

[38] S. Ueda, A. Iwasaki, M. Yokoo, M. Silaghi, K. Hirayama, and T. Matsui, "Coalition structure generation based on distributed constraint optimization," 2010.

[39] A. Petcu and B. Faltings, "Dpop: A scalable method for multiagent constraint optimization," in *IJCAI*, 2005, pp. 266–271.

[40] T. Voice, S. D. Ramchurn, and N. R. Jennings, "On coalition formation with sparse synergies," in *International Conference on Autonomous Agents and Multiagent Systems*, 2012, pp. 223–230.

[41] T. Voice, M. Polukarov, and N. Jennings, "Coalition structure generation over graphs," *Journal of Artificial Intelligence Research*, vol. 45, pp. 165–196, 2012.

[42] F. Bistaffa, A. Farinelli, J. Cerquides, J. Rodríguez-Aguilar, and S. D. Ramchurn, "Anytime coalition structure generation on synergy graphs," in *International Conference on Autonomous Agents and Multi-agent Systems*, 2014, pp. 13–20.

[43] T. Wang, L. Song, Z. Han, and W. Saad, "Distributed cooperative sensing in cognitive radio networks: An overlapping coalition formation approach," *IEEE Trans. Communications*, vol. 62, no. 9, pp. 3144–3160, 2014.

[44] W. Saad, Z. Han, M. Debbah, A. Hjørungnes, and T. Basar, "Coalitional games for distributed collaborative spectrum sensing in cognitive radio networks," in *IEEE INFOCOM*, 2009, pp. 2114–2122.

[45] Z. Zhang, L. Song, Z. Han, W. Saad, and Z. Lu, "Overlapping coalition formation games for cooperative interference management in small cell networks," in *IEEE WCNC*, 2013, pp. 643–648.

[46] W. Saad, Z. Han, T. Basar, M. Debbah, and A. Hjørungnes, "A selfish approach to coalition formation among unmanned air vehicles in wireless networks," in *International Conference on Game Theory for Networks*, 2009, pp. 259–267.

[47] D. Niyato, P. Wang, W. Saad, and A. Hjørungnes, "Coalition formation games for bandwidth sharing in vehicle-to-roadside communications," in *IEEE WCNC*, 2010, pp. 1–5.

[48] W. Saad, Z. Han, T. Basar, M. Debbah, and A. Hjørungnes, "A coalition formation game in partition form for peer-to-peer file sharing networks," in *IEEE GLOBECOM*, 2010, pp. 1–5.

[49] D. Niyato, Z. Han, W. Saad, and A. Hjørungnes, "A controlled coalitional game for wireless connection sharing and bandwidth allocation in mobile social networks," in *IEEE GLOBECOM*, 2010, pp. 1–5.

[50] R. M. Thrall and W. F. Lucas, "N-person games in partition function form," *Naval Research Logistics Quarterly*, vol. 10, no. 1, pp. 281–298, 1963.

[51] T. Michalak, A. Dowell, P. McBurney, and M. Wooldridge, "Optimal coalition structure generation in partition function games," in *European Conference on Artificial Intelligence*, 2008, pp. 388–392.

[52] S.-S. Yi, "Endogenous formation of economic coalitions: a survey of the partition function approach," in *The Endogenous Formation of Economic Coalitions*, 2003.