This paper was presented as part of the Mini-Conference at IEEE INFOCOM 2011

A Cost-Benefit Framework for Judicious Enterprise Network Redesign

Xin Sun and Sanjay G. Rao
Purdue University

***Abstract*—Recent works [1], [2] have shown the benefits of a systematic approach to designing enterprise networks. However, these works are limited to the design of greenfield (newly deployed) networks, or to incremental evolution of existing networks without altering prior design decisions. In this paper, we focus on redesigning existing networks, allowing for changes to existing decisions. Such redesign (migration) may be desirable from the perspective of improved network performance or lower complexity. However, the key challenge is that the costs of redesign may be high due to the presence of complex dependencies between network configurations. We consider these issues in the context of virtual local area networks (VLANs), an important area of enterprise network design.**

We make three contributions. First, we present a model to capture VLAN redesign costs. Such costs may arise from the need to reconfigure policies (e.g., security policies) to reflect the changes in VLAN design and ensure the continued correctness of the network. Second, we present a framework that enables operators to systematically determine the best strategies to redesign VLANs so the desired performance goals may be achieved while the costs of redesign are minimized. Finally, we demonstrate the effectiveness of our approach using data obtained from a large-scale campus network.

I. Introduction

Enterprise network operators must frequently change the design of their networks to reflect new organizational needs, that may arise due to the addition of new hosts, movement and reorganization of departments and personnel, revision of security policies, and upgrading of router hardware. Ad-hoc decisions made during the evolution of the networks may result in network designs that fall short of operator goals. For instance, they could result in VLANs with undesirably high broadcast traffic [3], routing designs with instabilities [4], [5], or unnecessarily complex network designs [6].

Redesigning enterprise networks could potentially lead to significant benefits such as improved performance or lowered complexity. However such benefits do not come without cost. The cost arises from the need to change multiple device configurations, to reflect the change in the design and ensure the continued correctness of the network. Reconfiguring networks is complicated given the huge semantic gap between high-level operator objectives, and the low-level configurations in which they are embedded, and given the presence of a large number of dependencies between configurations [6]. Errors in changing configurations have been known to result in outages, business service disruptions, violations of Service Level Agreements (SLA) and cyber-attacks [7], [8], [9].

Prior works including our own have looked at the systematic design of greenfield networks (i.e., networks yet to be deployed) [1], [10]. However, these works do not consider the redesign of existing networks. Our more recent work [2] has explored incremental evolution of existing networks. In particular, in [2], we developed strategies for dealing with routine enterprise change activity such as deciding which VLAN a newly added host must be assigned to. However, this work assumes prior design decisions may not be revisited. For instance, a host already assigned to a VLAN cannot be moved to another VLAN.

In this paper, we take a step towards tackling challenges encountered when existing enterprise networks are redesigned (allowing for prior decisions to be revisited). In particular, we present frameworks that enable operators to systematically determine how they should change the designs of their networks so they may achieve their objectives (e.g., improved performance), in a manner that incurs least reconfiguration effort. Our overall approach is to cast the network redesign process as an optimization problem that seeks to minimize the network reconfiguration costs subject to a performance goal. Our cost models capture network-wide dependencies associated with reconfiguring networks. We focus on Virtual LANs (VLANs) as a case study since they are extensively used in enterprise networks, represent time-consuming tasks for network operators, and have only recently started receiving attention from the research community [1], [2], [3]. The primary benefit of altering the VLAN design is that the broadcast traffic can be lowered. The costs incurred in reconfiguring VLANs involve considering dependencies such as changes of IP addresses of hosts, changes of security policies, and changes pertaining to which VLANs are permitted on particular links (a process referred to as changing "trunk" link configuration). We devise algorithms that can identify the set of design changes that incur minimum reconfiguration costs while achieving given

978-1-4244-9921-2/11/$26.00 ©2011 IEEE

performance goals. Finally, we evaluate our algorithms on a large-scale campus network, using a longitudinal data-set that includes multiple configuration snapshots. The results show promise of our approach.

II. Formulating the VLAN redesign problem

VLANs enable operators to reduce the management complexity by thinking about users as collective groups based on the role of each user in the organization. Hosts in an enterprise network may be viewed as corresponding to different roles or *categories*, e.g., engineering, sales, etc. Today, these logical groupings are most commonly implemented by VLANs, which take users in physically disparate locations and place them into a single logical subnet. Each VLAN is given its own IP block, typically a /24. Each VLAN constitutes a separate broadcast domain, and a separate spanning tree is constructed per VLAN, rooted at a root bridge.

Benefit of redesign: The benefits of redesigning VLANs come from the fact that the performance (defined in §II-A) of VLANs degrades significantly due to network evolution. Our recent work [1] targeted at greenfield settings showed that through more systematic assignment of hosts to VLANs, and judicious selection of root bridges, the broadcast traffic on core links of a large-scale operational network could be reduced by over 24%. Such inefficiency is due to changes in organizational needs over time. Typical changes are addition and removal of hosts, movement of hosts to different parts of the network, etc. As a result, many of the design decisions made earlier that were optimum then are no longer so now.

A. Considerations for VLAN redesign

We present the key considerations for redoing the VLAN design of networks:

Correctness criterion: Hosts in different categories are required to be placed in different VLANs. Note that this does not prohibit multiple VLANs being created for hosts in the same category, for example, to limit the broadcast traffic.

Resource constraints: The total number of VLANs in the network must be kept limited, as the demand on routers and switches grows with the number of VLANs. In this paper, we focus on a particular cost function like in [1], where the operator specifies an acceptable bound on the total number of VLANs.

Performance criterion: The key performance criterion we consider when redoing the VLAN design is the broadcast traffic associated with a VLAN. This in turn depends on (i) the number of hosts in the VLAN; and (ii) the span of the VLAN, i.e., how spread out the hosts of the VLAN are in the underlying network topology. The latter can be measured by the number of links in the spanning tree of the VLAN. More Formally, we leverage the broadcast traffic cost model from [1], which is defined as follows. Let N_i denote the number of hosts in a VLAN V_i, A_i denote the average broadcast traffic (in pkt/s) generated by a host in V_i, and W_{ik} denote the number of links in the spanning tree of V_i when it selects switch S_k as its root bridge. Then, the broadcast cost of V_i is modelled as:

$$BroadcastCost_i = N_i \times A_i \times W_{ik}$$

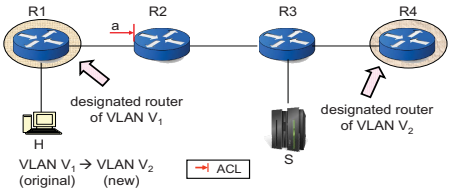

Fig. 1: Moving H from VLAN V_1 to V_2 results in the change of H's IP address and the path host H takes to reach server S.

B. Modeling redesign costs

We now proceed to consider ways to redo the VLAN design, and present a model for the associated reconfiguration costs. There are three types of changes that may occur during a VLAN redesign process:

- *Changing root-bridge of a VLAN:* The broadcast traffic associated with a VLAN depends on the choice of root-bridge, and could be lowered through a more judicious root-bridge placement. The main cost of changing the root-bridge is to reconfigure certain inter-switch links (a.k.a. trunk links) to ensure that the new root-bridge is part of the spanning tree. We note that each inter-switch link in a VLAN spanning tree must be explicitly configured to *only* permit traffic for the appropriate set of VLANs, to ensure that broadcast traffic is properly constrained. We denote the cost of changing the root-bridge of VLAN V_i to switch S_j to be $C(V_i, S_j)$.
- *Moving hosts across VLANs:* Hosts may be moved across VLANs (possibly to newly created VLANs) in the same category to ensure a more equitable distribution of hosts, and hence broadcast traffic across VLANs. We denote the cost of moving a host H_k to a different VLAN V_l to be $D(H_k, V_l)$. Below we describe in detail the configuration costs involved in moving a host.

First, reconfiguring (potentially multiple) switches to ensure the host is allowed to communicate with other hosts in the new VLAN, and disallowed from communication with the original VLAN.

Second, changing the IP address of the host to an IP that is within the address block assigned to the new VLAN. Changing a host's IP address in turn requires modification to DNS servers, routing (e.g., static routes),

packet filters and firewall rules on devices that are written in terms of the old IP address. Fig. 1 illustrates a scenario where a host H in VLAN V_1 is blocked from communicating with a server S by using an access-control list (ACL) a. An ACL is a sequential collection of permit and deny conditions, called ACL rules, and flows to be permitted/denied are identified in terms of their source/destination IP addresses and ports, and protocol.ACLs are placed on router interfaces. In our example, if during the redesign process host H is moved to another VLAN V_2, the rules in ACL a must be manually re-written to use the new IP address of H.

Third, moving hosts across VLANs may also change the forwarding path. This may imply that achieving the same security policy would require changing the devices over which security policies are configured. Consider Fig. 1 again where host H is moved from VLAN V_1 to V_2 as recommended by the redesign. Assume that V_1 and V_2 have *designated routers* R_1 and R_4, respectively (i.e., R_1 is the first (last) router for outgoing (incoming) packets when a host inside V_1 communicates with a host outside). Realizing this change now requires moving ACL a and ensuring that it is placed on the new path between H and S.

• *Creation or removal of VLANs:* The broadcast traffic associated with a VLAN could be lowered by creating a new VLAN and moving hosts to the new VLAN. Further, to ensure the total number of VLANs used in the design does not grow too large, VLANs with fewer hosts could be merged with larger VLANs in the same category. The associated configuration costs are primarily the costs in moving hosts to (from) the created (removed) VLAN. Hence, we do not explicitly list the costs associated with these tasks.

We next discuss how the cost functions C and D may be defined. In general, the reconfiguration costs must capture the dependencies across network devices that a configuration task entails. A potential cost model is to consider the number of devices whose configuration may need to be altered, and the number of different blocks of each configuration that must be changed. The more involved the change in these dimensions, the higher the degree of reconfiguration effort that is likely to be required and the larger the likelihood of errors in the process. While this is a reasonable approach, precisely determining the dependencies could be challenging, and is part of the complexity of the reconfiguration process. An alternative is a coarser model which assigns all operations of the same kind a uniform cost (e.g. $D(H_k, V_l) = D, \forall H_k, V_l$, where D is fixed). The cost of various types of operations could be weighed by the relative complexity involved with each of these tasks on average. The weights could be set in a variety of ways such as (i) assigning weights based on interviews with operators; or (ii) allowing the operator to weigh the relative costs associated with the tasks over-riding default models; or (iii) logging operator actions as they perform these tasks, and learning information about typical dependencies associated with the tasks.

C. Problem formulation

Given the maximum acceptable broadcast traffic cost B_T any VLAN in the network can have, probably specified by the operator, the VLAN redesign problem is to reduce the maximum broadcast traffic cost to B_T or below, while minimizing the redesign costs, subject to the correctness criterion and the resource constrains(§II-A):

Minimize:

$$\sum_i \sum_j C(V_i, S_j) x_{ij} + \sum_k \sum_l D(h_k, V_l) y_{kl}$$

Subject to:

- $\max_i \{\textit{BroadcastCost}_i\} \leq B_T$
- $\sum_j x_{ij} = 1, \ \sum_l y_{kl} = 1, \quad x_{ij}, y_{kl} \in \{0, 1\}$
- The correctness & resource criteria

Here, x_{ij} is 1 if VLAN V_i has changed root-bridge to switch S_j, and 0 otherwise. Similarly, y_{kl} is 1 if host H_k has changed its VLAN to be V_l, and 0 otherwise.

III. Solving the VLAN redesign problem

We first describe two important procedures, and then present the main algorithm.

A. Algorithm for moving a host: Move(V)

The $Move(V)$ procedure iteratively and greedily moves hosts from VLAN V to other VLANs in the same category, and/or changes the root-bridge of V, until V's broadcast traffic cost $B(V)$ has been reduced to B_T, or no more host can be moved and no better root-bridge exists. During each iteration, the algorithm either moves one host, or changes the root-bridge, depending on which action results in the maximum reduction of $B(V)$, and incurs the minimum redesign costs.

We now provide more details. First, for each host H in V, and for each VLAN V' in the same category of V, the algorithm calculates three variables: $D(H, V')$, the redesign costs of moving H to V_i; $Dec(H, V)$, the reduction in $B(V)$ if H is moved out; and $Inc(H, V')$, the increase in $B(V')$, if H is moved to V'. In addition, the algorithm also calculates the broadcast reduction $Dec(V, S)$ and the redesign costs $C(V, S)$ associated with selecting the optimal root-bridge S. The optimal root-bridge is determined by considering every single switch in the network as a potential root-bridge, and calculating the corresponding broadcast traffic cost. Next, the algorithm calculates the benefit-to-cost ratio

$\frac{Dec(H,V)/Inc(H,V')}{D(H,V')}$ for all the (H, V') pairs, as well as the ratio $\frac{Dec(V,S)}{C(V,S)}$. It then executes the redesign action that has the largest ratio, which can be either moving a host, or changing the root-bridge. The algorithm repeats the above steps, until $B(V)$ has been reduced to B_T, in which case it returns a "success", or no more host can be moved out from V without causing another VLAN's broadcast cost exceeding B_T, and no better root-bridge exists, in which case it returns a "failure", and undo all the previous actions.

B. Algorithm for merging a VLAN: Merge()

The $Merge()$ procedure merges *one* VLAN V to other VLANs in its category C. The algorithm judiciously picks the VLAN V such that (i) the redesign costs associated with merging V, which is the sum of the costs of moving out all the hosts in V, is smaller than merging any other VLAN in the network; and (ii) the merge will not cause any other VLAN's broadcast cost to exceed B_T.

Below is the detailed description of the algorithm. The algorithm considers all the categories with at least two VLANs. For each of those categories C, and for each VLAN V in C, it calculates the redesign costs of moving out all the hosts in V, by calling a procedure $P(V)$. $P(V)$ is similar to the *Move(V)* procedure presented above, except that it does not consider changing the root-bridge, and that it returns a "success" only when having moved out all hosts in V. Procedure $P(V)$ ensures that moving out all the hosts of V will not cause any other VLAN's broadcast traffic cost to exceed B_T. If that is not feasible (i.e., $P(V)$ returns a "failure"), V will not be considered for merge. The algorithm then picks V for which the merging costs are the smallest, executes the merge, and returns a "success". If no such V available, the algorithm returns a "failure".

C. The main algorithm

Finally we describe the main algorithm. The algorithm iterates over every VLAN V whose broadcast traffic cost $B(V)$ is greater than B_T. For each such V, the algorithm calls the $Move(V)$ procedure to move hosts out of V and/or change the root-bridge of V. If $Move(V)$ returns a "success", which means $B(V)$ has been reduced to B_T, the algorithm finishes the current iteration, and goes on to the next VLAN. Otherwise if $Move(V)$ returns a "failure", which means it is not feasible to reduce $B(V)$ to below B_T without causing another VLAN's broadcast traffic cost to go above B_T, the algorithm will try to create a new VLAN in the same category as V. Depending on whether the total number of VLANs currently employed is smaller than *MAX-VLANs* or not, the algorithm either creates a new VLAN directly, or first calls the $Merge()$ procedure to merge a VLAN in another category, and then creates a new VLAN. The algorithm then calls $Move(V)$ again.

	09/2008	03/2009	09/2009
Current max (pkt/s)	226700	225085	228854
Cost-agnostic max (pkt/s)	84223	83079	76782

TABLE I: Performance comparison of the current VLAN design, and the cost-agnostic VLAN design, for the three snapshots.

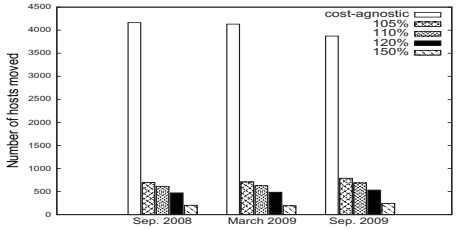

Fig. 2: Number of hosts that are moved by the cost-agnostic algorithm, and by our algorithms with various B_T values.

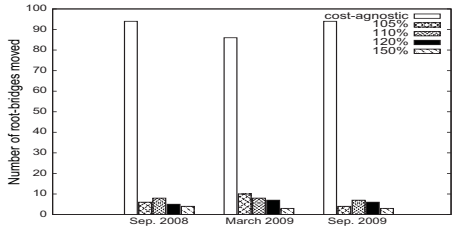

Fig. 3: Number of root-bridges that are changed by the cost-agnostic algorithm, and by our algorithms with various B_T values.

IV. Evaluation

In this section, we evaluate the benefit of our redesign algorithms. Our data-set includes multiple snapshots of the configuration files of all switches and routers in a large-scale campus network, as well as layer two topology data, from 2007 to 2009. The network consists of about 200 routers, 1300 switches, and tens of thousands of hosts grouped into hundreds of VLANs.

We pick three representative snapshots of the campus network from our data-set, each six months apart from another, and run our redesign algorithms on them. For comparison purpose, we also run a ***cost-agnostic*** design algorithm that we previously developed [1] for greenfield networks (i.e., networks that are yet to be deployed). It does not take into account the configuration of the current network, or the redesign costs, and comes up with a grouping of hosts into VLANs with the only objective being to ensure the broadcast traffic of each VLAN is as low as possible. Thus the result produced by this algorithm may be viewed as an upper bound of the performance that the network could achieve through redesign.

We assume every host generates broadcast traffic at a rate of 2.12 packet per second, based on a measurement study reported in [1]. We assume the maximum number of VLANs that can be created is the number in the current design. We use a uniform cost model, and the same cost is assigned to an operation involving moving any host to any VLAN, or changing the root-bridge of any VLAN. We believe this model is reasonable, as discussed in §II-B. Finally, we vary the maximum acceptable broadcast traffic cost (B_T) to be between 105% and 150% of the maximum broadcast traffic cost produced by the cost-agnostic design.

Table I shows the the *maximum* broadcast traffic generated by the current design, and by the cost-agnostic design. Each column in the table corresponds to a different snapshot. For all the snapshots, the current design incurs significantly larger broadcast traffic cost than the cost-agnostic design.

Fig. 2 shows the results regarding the number of hosts moved. There are three sets of bars, each corresponding to a different snapshot. In each set, there are five bars. The leftmost bar shows the number of hosts moved by the cost-agnostic design, and the rest bars show the number of hosts moved by our algorithms with different B_T values. We make the following two observations. First, the cost-agnostic design requires moving about 4000 hosts, which is clearly not practical. This is because the cost-agnostic design solely optimizes for performance, and it does not take into account the redesign costs. Second, our algorithms can achieve comparable performance to the cost-agnostic designs, while moving significantly fewer hosts. For example, to reduce the maximum broadcast traffic cost to within 150% of that of the cost-agnostic design, our algorithms moved about 200 hosts, only 5% of the hosts moved by the cost-agnostic design.

Fig. 3 shows similar trends regarding the number of root-bridges changed. For all the snapshots, the cost-agnostic design requires around 90 root-bridges to be changed, while our algorithms require less than 10.

V. Related work and conclusion

In this paper, we have made three contributions. First, we present a model to capture VLAN redesign costs. Second, we present a framework that enables operators to systematically determine the best strategies to redesign the VLANs so the desired performance goals may be achieved while the costs of redesign is minimized. Finally, we demonstrate the effectiveness of our approach using data obtained from a large-scale campus network.

Prior work on systematic "top-down" approaches to enterprise design and configuration is in the limited context of newly deployed (greenfield) networks [1], [10]. Such approaches cannot be used to make redesign recommendations, as they are completely cost agnostic and will incur prohibitively high costs, as shown in §IV. In contrast, our approach explicitly considers various costs associated with redesign. Researchers have proposed new architectures that provide alternatives to broadcast-based host discovery [11], [12] potentially obviating the need to constrain broadcast domains with VLANs. In contrast, our focus is on tackling the challenges within the constraints of existing enterprise network architectures. [13] presents techniques for simplifying configuration by reorganizing policies. In contrast, our focus is on improving network performance through judicious redesign. Finally, while this paper focuses on VLANs as a case study, we believe a framework such as ours could be important in other aspects of enterprise networks, such as routing design. We leave further investigation of these issues for future work.

Acknowledgment

We thank Brad Devine, Duane Kyburz and other colleagues in the Information Technology Department at Purdue (ITaP) for providing access to the data, and for being generous with their time. This material is based upon work supported by the National Science Foundation under Grant No. 0953622.

References

[1] Y.-W. E. Sung, S. G. Rao, G. G. Xie, and D. A. Maltz, "Towards systematic design of enterprise networks," in *Proc. CoNEXT*, 2008.

[2] X. Sun, Y.-W. E. Sung, S. Krothapalli, and S. Rao, "A Systematic Approach for Evolving VLAN Design," in *Proc. IEEE INFOCOM*, 2010.

[3] P. Garimella, Y.-W. E. Sung, N. Zhang, and S. Rao, "Characterizing VLAN usage in an operational network," in *Proc. SIGCOMM INM Workshop*, 2007.

[4] F. Le, G. G. Xie, D. Pei, J. Wang, and H. Zhang, "Shedding light on the glue logic of the Internet routing architecture," in *Proc. SIGCOMM*, 2008.

[5] D. Maltz, G. Xie, J. Zhan, H. Zhang, G. Hjalmtysson, and A. Greenberg, "Routing design in operational networks: A look from the inside," in *Proc. ACM SIGCOMM*, Aug. 2004.

[6] T. Benson, A. Akella, and D. Maltz, "Unraveling the complexity of network management," in *Proc. NSDI*, 2009.

[7] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding BGP misconfiguration," in *Proc. ACM SIGCOMM*, Aug. 2002.

[8] Z. Kerravala, "Configuration management delivers business resiliency," The Yankee Group, Nov. 2002.

[9] S. Narain, "Network configuration management via model finding," in *Proc. LISA*, Dec. 2005.

[10] W. Enck, P. McDaniel, S. Sen, P. Sebos, S. Spoerel, A. Greenberg, S. Rao, and W. Aiello, "Configuration management at massive scale: System design and experience," in *Proc. of USENIX Annual Technical Conference*, January 2007.

[11] C. Kim, M. Caesar, and J. Rexford, "Floodless in SEATTLE: A scalable Ethernet architecture for large enterprises," in *Proc. SIGCOMM*, 2008.

[12] A. Greenberg, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "VL2: A scalable and flexible data center network," in *Proc. SIGCOMM*, 2009.

[13] S. Lee, T. Wong, and H. S. Kim, "Improving dependability of network configuration through policy classification," in *Proc. of DSN*, 2008.