This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Towards Systematic Design of Enterprise Networks

Yu-Wei Eric Sung, Xin Sun, Sanjay G. Rao, Geoffrey G. Xie, and David A. Maltz

***Abstract*—Enterprise networks are important, with size and complexity even surpassing carrier networks. Yet, the design of enterprise networks remains *ad hoc* and poorly understood. In this paper, we show how a systematic design approach can handle two key areas of enterprise design: virtual local area networks (VLANs) and reachability control. We focus on these tasks given their complexity, prevalence, and time-consuming nature. Our contributions are threefold. First, we show how these design tasks may be formulated in terms of network-wide performance, security, and resilience requirements. Our formulations capture the correctness and feasibility constraints on the design, and they model each task as one of optimizing desired criteria subject to the constraints. The optimization criteria may further be customized to meet operator-preferred design strategies. Second, we develop a set of algorithms to solve the problems that we formulate. Third, we demonstrate the feasibility and value of our systematic design approach through validation on a large-scale campus network with hundreds of routers and VLANs.**

***Index Terms*—Configuration management, enterprise networks, network design, security policies.**

I. Introduction

RECENT empirical studies reveal that the size of some enterprise networks and the complexity of their routing designs rival or even surpass those of carrier networks [1], [2]. Far more enterprise networks than carrier networks are in operation today, and their designs are highly customized to the needs of individual companies, universities, government agencies, or other types of organizations. However, despite their complexity, prevalence, and diversity, enterprise networks have received little attention from the research community.

Managers of enterprise networks face unique design challenges. They need to meet a wider range of security, resilience, and performance requirements than managers of carrier networks. Examples of such challenges include the configuration of virtual local area networks (VLANs) to ease the management of different user groups [3], the integration of multiple routing domains to support company mergers [2], and the installation of packet filters to perform ingress filtering and to control access to privileged databases [4].

The unique challenges of enterprise network design have further exposed the limitations of the existing *ad hoc* approach to network design and management. On the one hand, a manager faces high-level constraints such as performance, ease of manageability, security, and resilience to failures. On the other hand, to realize a network design, a manager must manually choose from a slew of protocols, low-level mechanisms, and options. Many of these protocols and mechanisms have profound interactions. However, the current "protocol by protocol" method of network configuration does not allow the network operator to see and control these interactions in a systematic manner. Design faults and configuration errors account for a substantial number of network problems [5] and are exploited by over 65% of cyber-attacks according to recent statistics [6].

In this paper, we explore the feasibility of adopting a systematic approach to enterprise network design. The key elements include: 1) identifying the network-wide performance, security, and resilience requirements of a task; 2) formulating the requirements as one of optimizing desired (operator-customized) criteria subject to correctness and feasibility constraints on the design; and 3) developing algorithms and heuristics to solve the formulated problems.

We show that two critical enterprise network design tasks lend themselves to such a systematic approach. These include: 1) VLAN design; and 2) reachability control through placement of packet filters. We model the objectives of VLAN design as achieving low costs associated with broadcast and data traffic, given constraints such as a categorization of hosts into distinct logical groups and a limit on the number of VLANs used. We model the objectives of packet filter placement as optimizing for operator-specified placement criteria such as balancing processing needs across routers, while correctly realizing desired security policies, and meeting feasibility constraints on the processing capacities of routers.

We evaluate the benefits of a systematic design approach in the context of algorithms we developed to solve our formulated problems. Our validations are conducted on a large-scale campus network data set involving hundreds of routers and VLANs and a few thousand switches. Beyond the general time savings in realizing a correct and easily customizable design, our results show that through systematic VLAN design, broadcast and data traffic can be reduced by over 24% and 55%, respectively. Our results also highlight the importance of a systematic approach to placing packet filters by identifying inconsistencies in the realization of operator security objectives in the campus network data set. Overall, these results show the promise of a systematic design approach in these key areas and are a first but key step toward the top-down design of enterprise networks in general.

Manuscript received June 25, 2009; revised July 20, 2010; accepted September 27, 2010; approved by IEEE/ACM Transactions on Networking Editor M. Kodialam. This work was supported by NSF Awards CNS-0721488, CNS-0520210, and CNS-0721574.

Y.-W. E. Sung was with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA. He is now with Facebook, Palo Alto, CA 94304 USA (e-mail: sungy@ecn.purdue.edu).

X. Sun and S. G. Rao are with the Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: sun19@purdue.edu; sanjay@ecn.purdue.edu).

G. G. Xie is with the Department of Computer Science, Naval Postgraduate School, Monterey, CA 93943 USA (e-mail: xie@nps.edu).

D. A. Maltz is with the Networking Group, Microsoft Research, Seattle, WA 98052 USA (e-mail: dmaltz@microsoft.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2010.2089640

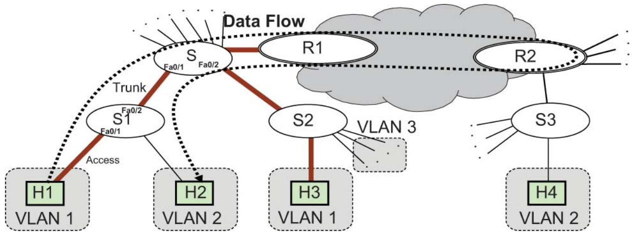

Fig. 1. Communication between different VLANs is routed through designated routers. When H1 talks to H2, R1 acts as a router in the outgoing direction, but as a switch in the return direction.

II. Framing Enterprise Design Tasks

The nature of the enterprise design problem is little known outside the operational community. For example, there is almost no coverage of this topic in college textbooks. Only through repeated inspections of router configuration files and close interactions with network managers have we obtained a basic understanding of what technical challenges it entails.

We observe that enterprise design can be decomposed into a sequence of distinct stages or tasks. The major tasks in order of execution are: 1) plan physical topology and wiring; 2) create VLANs and layer-2 topology; 3) select and configure routing protocols; and 4) control reachability with packet filters or firewalls. This work focuses on tasks 1) and 4) because these tasks have been identified by network managers as challenging and time-consuming, and they have been relatively unexplored by the research community. In the rest of this section, we give a high-level description of the technical challenges facing VLAN design and reachability control.

A. VLAN Design

Operators reduce the complexity of their configuration tasks by thinking about users as collective groups based on the role of each user in the organization (e.g., what resources they should be able to access). Today, these groupings are most commonly implemented by VLANs, which take a set of users in physically disparate locations and place them into a single logical subnet, even if the users are connected to different switches. For instance, an enterprise policy may permit Web access only for all sales personnel, and it may be desirable to ensure these users receive IP addresses from the same subnet so that routing policies and packet filters can be applied to them as a group. Consider Fig. 1. S and S1–S3 are switches, and R1–R2 are routers. Notice that even though hosts H1 and H3 are physically separated, they are both part of VLAN 1. Likewise, hosts H2 and H4 belong to VLAN 2.

Each VLAN constitutes a separate broadcast domain. Therefore, it is important to ensure that broadcast traffic is properly constrained to reduce unnecessary traffic for increased performance and security. To achieve this, every link is configured to permit only traffic for appropriate VLANs. In Fig. 1, the link S1–H1 is configured as an *access* link and forwards only VLAN 1 traffic. The link S1–S is configured as a *trunk* link and permits traffic for multiple explicitly specified VLANs (in this case, VLANs 1 and 2). Typically, a separate spanning tree rooted at a *root bridge* is constructed per VLAN. For example, the collection of bold links forms the spanning tree of VLAN 1, with S being its root bridge.

Each publicly accessible VLAN is assigned with what we term a *designated (gateway) router* for that VLAN. When a host inside a VLAN communicates with a host outside, the designated router is the first (last) router for outgoing (incoming) packets. In Fig. 1, R1 and R2 are respectively the designated routers for VLAN 1 and VLAN 2. The IP level path between H1 and H2 is $H1$–$R1 \ldots R2$–$H2$, with $R1 \ldots R2$ denoting there could be other routers in the path. The path of data flow is also highlighted in the figure.

In VLAN design, an operator faces two key tasks with unique technical challenges.

1) **Grouping hosts into VLANs:** The operator must decide the appropriate number of VLANs in the design and determine which hosts must belong to each VLAN. In doing so, three factors must be considered. First, security policies and management objectives may influence the decision. For example, in a campus network, the manager may desire to separate faculty and student machines into different VLANs in order to provide faculty with greater access to servers hosting confidential documents. Second, hosts in a VLAN belong to the same broadcast domain, and it is important to keep the cost of broadcast traffic small. The cost depends both on: a) the number of hosts in the VLAN; and b) the span of the VLAN, i.e., how spread out the hosts of the VLAN are in the underlying network topology. Finally, the total number of VLANs in the network must be kept limited, as the demand on network hardware grows with the number of VLANs. For instance, a separate spanning tree is typically constructed and maintained for every VLAN in the network, and this increases the memory and processing requirements of individual switches.

2) **Placement of router and bridge:** For each VLAN with the host assignment decided, the operator must determine the best locations of the designated router and the root bridge of the spanning tree. A key consideration is the potential inefficiencies in data communication with VLANs. Consider Fig. 1. Even though H1 and H2 are physically connected to the same switch, the path along which data flows is substantially longer. Having longer paths not only leads to longer delays, but also increases the likelihood of failures and complicates performance and failure diagnosis. For example, if H1 and H2 were in building X, and R2 were in building Y, communication could be disrupted by a power failure in a building located between X and Y.

   The inefficiencies of communication between H1 and H2 would be reduced if R1 were chosen as the designated router of VLAN 2 instead of R2. An ideal placement strategy must consider both the location of all the hosts in the VLAN and the traffic patterns of the hosts. For instance, if hosts in a VLAN tend to communicate with certain servers, it is more critical to limit the performance inefficiencies associated with communication involving those servers.

   The placement of root bridge directly impacts the spanning tree constructed for a VLAN. This in turn determines: 1) the network links that see broadcast traffic of the VLAN;

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

and 2) the hops traversed when a host in the VLAN communicates with its gateway router. Thus, it is important to place the root bridge judiciously to lower broadcast traffic in the network and reduce inefficiencies in data communication.

B. Reachability Control

From an operator's point of view, a primary objective of network security is to control packet-level reachability, that is, what packets sent by a traffic source are permitted to reach a destination. Common security policies, such as restricting the types of external applications a host can access, limiting the scope of multicast traffic to specific subnets, and blocking unauthorized ICMP and SNMP probes, are essentially about permitting packets with particular header field combinations to be exchanged between hosts. Current design approaches are *ad hoc* and error-prone, and current best practices for validating if a network configuration meets given reachability control objectives involve *in situ* testing [4].

Today, operators realize reachability control objectives using two configuration options. The first is a data plane solution, which installs access control lists (ACLs), also commonly referred to as packet filters, on router interfaces. An ACL is a sequential collection of permit and deny conditions, called ACL rules. A packet's header fields are matched against each rule successively. The order of rules is critical because testing stops with the *first match*. If no match is found, an implicit default rule is assumed. In many cases, the rule is "deny any," thereby rejecting all unmatched packets.

The second approach to achieving reachability control objectives is a control plane solution. In particular, by either depriving some routers of certain routes or creating black-hole routes in their forwarding tables, unwanted packets may be dropped by the routing logic. For example, one may partition a network into multiple routing domains and restrict the flow of routing information between the domains so that not all routers have routes to all destinations in the network.

Controlling reachability through the routing design has a much smaller CPU overhead because the execution of routing logic, particularly the lookup of the forwarding table, is mostly performed by forwarding hardware and requires little router CPU time. However, the routing-oriented solution is not always applicable because of its relatively limited range of conditions for matching packets. Unlike an ACL rule, which may simultaneously refer to multiple header fields, the routing logic matches packets either entirely based on source address or entirely on destination address.

Fig. 2 shows an example scenario where either configuration option can be used to meet a security policy. A1, A2, B1, B2, and C are subnets. Suppose the security policy does not permit any host in A2 and B2 to talk to C, but permits every host in A1 and B1 to talk to C. To realize this policy, the operator may configure an ACL, as shown in Fig. 2, in the inbound direction of both interfaces of router X2. Alternatively, the operator may block traffic between A2 and C, and between B2 and C, through routing design—one possible option is to install two source-address-based black-hole routes for traffic originated from A2 or B2 at router X2.

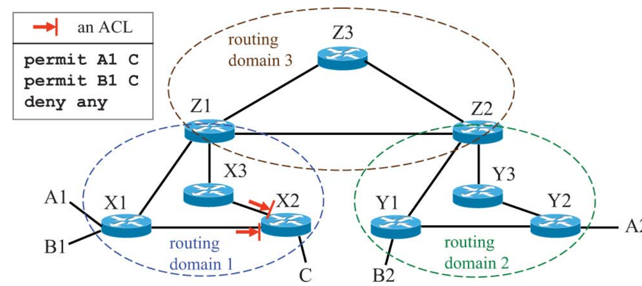

Fig. 2. Reachability control at data plane and control plane.

While routing design has been extensively studied (e.g., [7]–[9]), ACL placement has received little attention to date. In this paper, we focus on ACL placement. We assume that routing design is already completed, and routing domains are successfully configured before the operators proceed to determine the placement of ACLs in the network.

The key task with ACL placement is that operators need to construct a set of ACLs based on the security objectives and determine suitable locations, i.e., combinations of router interface and traffic direction, to place them. In coming up with an ACL placement, the primary criterion is **correctness** of the design. The ACL and routing configurations must guarantee the delivery of all authorized packets while preventing all unauthorized traffic from reaching the destination. The solution should also be resilient to certain link or router failure scenarios—in particular, the alternate paths that may be taken when failures occur must also be correctly configured to ensure the reachability constraints are always met.

Another consideration in ACL placement is the CPU overhead that routers incur from processing ACL rules *packet by packet*. There is a limit on the total number of ACL rules that a router can process consistently per packet. The limit varies from model to model. A low-end router may only be able to process dozens of ACL rules per packet without a noticeable reduction in link utilization. Therefore in some scenarios, it may be necessary to place ACLs throughout the network to distribute the computation cost. A recent study [1] reveals that some operational networks indeed have many ACLs placed at internal routers, in addition to ACLs placed at border routers.

III. Systematic VLAN Design

In this section, we present our approach for systematic VLAN design. We first describe the network-wide abstractions that capture the key aspects of VLAN design. We then formulate the VLAN design task as a two-phase process: 1) grouping hosts into VLANs; and 2) choosing the router and root bridge for each VLAN. For each phase, we present a problem formulation and then our solution. In Table I, we list the important notations that we use in the paper.

A. Network-Wide Abstractions

We consider the following abstractions.

- **Host category:** This is a mapping $\mathcal{P}$ that associates each host H_i in the network with the logical category $\mathcal{P}(H_i)$ (e.g., engineering, sales) to which it belongs. While hosts in the same category need not belong to the same VLAN,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
NOTATION TABLE

$\mathbf{M_T}$	inter-host traffic matrix	V_I	VLAN representing communications with the Internet
V_i	VLAN i		
H_i	host i	R_G	gateway router to the Internet
S_i	switch i	x_{ik}	indicator variable, 1 iff V_i's designated router is R_k
R_i	router i		
$\mathcal{P}(H_i)$	category of host H_i	d_{kl}	# of hops between R_k and R_l
$C(x)$	costs of having x VLANs		
N_i	# of hosts in V_i	d^*_{ik}	average # of hops between hosts in V_i and R_k
B_i	average broadcast traffic generated by a host in V_i	f_{ij}	data traffic from V_i to V_j
W_{ik}	# of links in V_i when S_k is the root bridge	RS	reachability set
		$\mathbf{M_R}$	reachability matrix
$Br(V_i, S_k)$	broadcast cost of V_i when S_k is the root bridge	$b(r)$	# of ACL rules configured on router r
$Br(V_i)$	broadcast cost of V_i with the optimum root bridge chosen	$c(r)$	maximum # of ACL rules allowed on router r

hosts in two different categories must belong to two different VLANs. This is the *correctness criterion* for VLAN design.

- **Traffic matrix:** This is a matrix $\mathbf{M_T}$ that specifies expected traffic patterns between hosts in two different categories (or same category, or a given category and the Internet). We assume information is provided about the *average* traffic between all host pairs in two categories. That is, $\mathbf{M_T}(i,j)$ specifies the average data traffic (in kilobits per second) sent by a host in category i to a host in category j. While a precise traffic matrix might be hard to obtain, we discuss in Sections III-C.3 how to work with coarse traffic patterns if accurate information is unavailable.
- **Network topology:** We abstract the network topology as a graph $\mathcal{G} = (\mathcal{H} \bigcup \mathcal{S}, \mathcal{E})$. $\mathcal{H} = \{H_1, H_2, \ldots H_h\}$ denotes the set of end-hosts. $\mathcal{S} = \{S_1, S_1, \ldots, S_s\}$ denotes the set of switches, i.e., devices that are capable of performing layer-2 switching. We let $\mathcal{R} = \{R_1, R_2, \ldots, R_r\} \subseteq \mathcal{S}$ denote the set of routers, i.e., the subset of switches that are also capable of performing layer-3 routing. $\mathcal{E}$ denotes the set of edges. Two vertices are connected by an edge in $\mathcal{G}$ if they are physically connected in the underlying network.

B. Phase 1: Grouping Hosts Into VLANs

We present a problem formulation for the host grouping problem, discuss the complexity, and present our solution.

1) Problem Formulation: Formally, consider the network $\mathcal{G} = (\mathcal{H} \bigcup \mathcal{S}, \mathcal{E})$ as defined in Section III-A. Let $C(x)$ denote the costs associated with having x VLANs in the design. Let $\mathcal{H}_1 \subseteq \mathcal{H}$ be a subset of hosts, and let $Br(\mathcal{H}_1, S_k)$ denote the broadcast costs associated with the VLAN that consists of all hosts in $\mathcal{H}_1$ and root bridge S_k. Let $Br(\mathcal{H}_1) = \min\{Br(\mathcal{H}_1, S_k),\ 1 \le k \le s\}$ denote the broadcast cost associated with the VLAN $\mathcal{H}_1$ for the best possible choice of root bridge.

Then, the VLAN grouping problem is to determine a partition $\mathcal{V} = \{V_1, V_2, \ldots, V_x\}$ of $\mathcal{H}$ such that

$$\text{Minimize} \quad C(x) + \max\{Br(V_i), 1 \le i \le x\} \tag{1}$$

$$\text{subject to:} \quad \forall V_i, H_1, H_2 \in V_i \Rightarrow \mathcal{P}(H_i) = \mathcal{P}(H_j). \tag{2}$$

In this paper, we focus on specific models for $C(x)$ and $Br(V_i, S_k)$, which we present next.

Costs Associated With Adding VLANs: We focus on a particular cost function, where the manager specifies an acceptable bound on the total number of VLANs. In particular, if x VLANs are employed in the design, and MAX-VLANs is the maximum number of VLANs acceptable in the design (a constraint provided by the manager, and probably derived from the number of VLANs supported by the routers and switches being used), then

$$C(x) = 0, \text{ if } x \le \text{MAX-VLANs} \tag{3}$$

$$C(x) = \infty, \text{ if } x > \text{MAX-VLANs}. \tag{4}$$

We believe this is a natural cost function that is easy to express to the operator and translates to many real-world design scenarios. While our current model may also be viewed as a feasibility criterion, it may be interesting to consider other kinds of cost functions in the future.

Broadcast Traffic Costs: Several applications may result in broadcast traffic in a network such as ARP, IPX, NetBIOS, DHCP, MS-SQL, etc. We model the broadcast traffic cost based on: 1) the rate at which broadcast traffic is generated; and 2) the number of links traversed as part of the broadcast. The links traversed by the broadcast traffic in a VLAN are simply the links present in the spanning tree for that VLAN. This may be easily generalized to a weighted sum of links, where weights are assigned to individual links to capture the cost of traversing that link.

In general, let N_i denote the number of hosts in VLAN V_i, B_i denote the average broadcast traffic (in kilobits per second) generated by a host in V_i, and also let W_{ik} denote the number of links in the spanning tree for V_i when S_k is chosen as its root bridge. Then, we model the broadcast cost for VLAN V_i as

$$\text{Br}(V_i, S_k) = N_i \times B_i \times W_{ik}. \tag{5}$$

We believe a linear dependence on the number of hosts in the network is a reasonable model. For instance, consider ARP queries, a key source of broadcast traffic. In typical scenarios, most ARP queries are sent by hosts in the VLAN for its designated router, or by the designated router for hosts in the VLAN, and a linear model fits well. Other models may be more appropriate in certain scenarios. For example, the entire IP address space of the VLAN may need to be considered for ARP broadcast storms due to port scans to nonexistent hosts in the VLAN. As another example, a quadratic model is more appropriate if there is significant intra-VLAN ARP traffic. These scenarios are less typical, but we believe it is easy to extend our model to consider them.

Computing the number of links W_{ik} in the spanning tree of the VLAN depends on where the router and root bridge are located, which are themselves unknowns, and a degree of freedom the manager enjoys. When partitioning hosts into VLANs, our solver assumes the router and root bridge are placed in a manner that would result in the smallest number of links in the spanning tree. Thus, host grouping indicates the feasibility of keeping the broadcast costs small subject to appropriate router and bridge placement. The second phase of the solver (Section III-C) determines router and bridge placement, with the broadcast traffic costs being one of the criteria.

2) Complexity of Problem:

Theorem III-1: The host grouping problem is NP-hard with respect to the number of hosts to be grouped.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

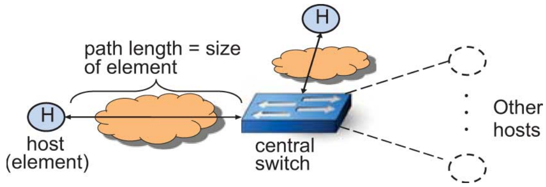

Fig. 3. Reduction from 3-partition problem to VLAN grouping problem.

Proof: Given our cost models, the problem of grouping hosts into VLANs involves minimizing the maximum broadcast cost across all VLANs subject to category constraints, with the broadcast cost defined as in (5). We consider a decision version of the problem, where the goal is to determine if a grouping exists such that all VLANs have a broadcast cost less than X, for a given X. We show this problem is NP-hard using a reduction from the well-known 3-partition problem, which is known to be NP-hard [10]. In the 3-partition problem, we are given a set $\mathcal{A}$ of $3m$ elements, with each $a \in \mathcal{A}$ associated with an integer size $s(a)$. Furthermore, $\sum_{a\in\mathcal{A}} s(a) = m \times B$, and $\forall a \in \mathcal{A}, B/4 < s(a) < B/2$. The problem is to decide if the set $\mathcal{A}$ can be partitioned into m subsets such that the sum of the size of the objects in each subset is identical (or exactly B). Note that since $\forall a \in \mathcal{A}, B/4 < s(a) < B/2$, each subset is forced to consist of exactly three elements.

To show the reduction, we consider a special version of the VLAN grouping problem for each instance of the 3-partition problem. In particular, we consider a topology as shown in Fig. 3. In this topology, there is a single central switch, and for each element in the 3-partition problem, we introduce a host that is connected to the central switch using a path of length equal to the size of the element. Furthermore, all hosts are assumed to belong to the same category and produce the same amount of broadcast traffic corresponding to 1 unit (e.g., 1 kb/s). We note that for any VLAN involving two or more hosts in this topology, the spanning tree must consist of the central switch, and all switches on the path from the central switch to each of the hosts. Thus, the number of links in the spanning tree is simply the sum of the path lengths of each host to the central switch, or equivalently, the sum of the sizes of the corresponding elements in the original 3-partition problem.

We claim that a feasible 3-partition exists in the original problem if and only if the decision version of the VLAN grouping problem returns true for m permitted VLANs with a bound on broadcast cost of $3 \times B$.

The proof of this claim has two parts.

- Let us assume a feasible 3-partition exists. Then, for all the elements mapped to one subset, we take the corresponding hosts and group them in one VLAN. Since the sum of elements in each subset is exactly B, and each subset has exactly three elements, the number of spanning tree links in each VLAN is B, and each VLAN has three hosts. Hence, the broadcast traffic is $3 \times B$ for each VLAN, and the decision version of the VLAN grouping problem returns true for the bound of $3 \times B$.
- Next, assume that the decision version of the VLAN grouping problem returns true for bound $3 \times B$, i.e., we can group hosts into VLANs such that each VLAN has broadcast traffic of at most $3 \times B$. We first show all VLANs must have exactly three hosts. If this were not the case, there must be some VLAN with four or more hosts (as there are $3m$ hosts to be partitioned into m VLANs). However, each host has a distance $> B/4$ from the central switch (as each element in original problem has size $> B/4$). Hence, for this VLAN, the number of links in the spanning tree is $> B$. Since the number of hosts ≥ 4, the broadcast cost for the VLAN is $> 4B$. This is a contradiction to our assumption and is not feasible. Given all m VLANs have exactly three hosts, the number of links in the spanning tree of each VLAN must be $\leq B$, as the maximum broadcast cost is $3 \times B$ across all VLANs. However, the sum of the number of links in spanning trees of all VLANs must be equal to the sum of the sizes of all elements in original problem $(m \times B)$. This is only possible if the number of links in spanning tree of every VLAN is exactly B. This means an algorithm for solving the VLAN grouping problem can also be used to solve the 3-partition problem, where we create m subsets, with each subset corresponding to a VLAN, and elements in that subset corresponding to hosts in that VLAN. ■

3) Heuristic for Creating Host Groupings: Given the complexity of the problem and the scale of enterprise networks, it is impractical for any algorithm to find out the optimum grouping. Instead, our solver employs a greedy heuristic to determine grouping of hosts into VLANs. Initially, each category of hosts provided by the operator is assumed to constitute one VLAN. The solver then computes the minimum broadcast traffic costs for each VLAN. The VLAN with the largest broadcast traffic cost is taken and is split into two VLANs if the total number of VLANs in the design is no more than MAX-VLANs. The process continues iteratively until the condition is violated.

When a VLAN V_i is chosen to be split, then the goal is to split it in a manner that hosts close to one another in the underlying topology are placed in one VLAN to minimize the span. The solver employs the following two-step algorithm:

1) For each host H_k in VLAN V_i, we compute the shortest distances from H_k to all N_i hosts in VLAN V_i, including itself, to form a vector $\{d(H_k, H_h) | h = 1 \ldots N_i \text{ and } H_h \in V_i\}$ of N_i values, where $d(H_k, H_h)$ denotes the shortest distance (i.e., number of layer-2 hops) from host H_k to host H_h in V_i.
2) Using the vector of a host as its coordinate (or location) in the topology, we perform the k-means algorithm to cluster all hosts in VLAN V_i into two separate VLANs.

C. Phase 2: Router and Bridge Placement

Once the hosts are grouped into VLANs, the placement of the designated router and the root bridge must be determined for each VLAN. We next present a problem formulation for the placement problem, discuss the complexity, and present our solution.

1) Problem Formulation: The key objective of the placement problem is minimizing the *combined* costs of data and broadcast traffic. The broadcast traffic cost was formulated in (5). The data traffic cost depends on two factors: 1) the amount of data traffic exchanged between a pair of hosts; and 2) the number of hops (switches and routers) traversed as part of the communication. In this paper, we focus on the scenario where the designated router and the root bridge for a VLAN are always coupled. This

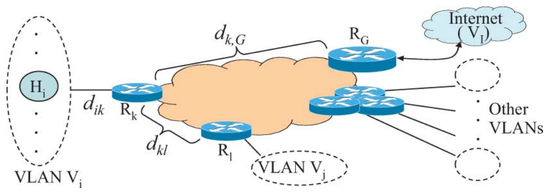

Fig. 4. Inter-VLAN traffic sent by a host in VLAN V_i.

is often the case in enterprise networks, as it simplifies the management tasks. We believe that our model can be easily extended to scenarios where the two are decoupled.

Formally, consider the network $\mathcal{G} = (\mathcal{H} \bigcup \mathcal{S}, \mathcal{E})$ as defined in Section III-A, with $\mathcal{R} \subseteq \mathcal{S}$ to denote the set of routers. Let $\mathcal{V} = \{V_1, V_2, \ldots, V_v\}$ be a partition of $\mathcal{H}$ into VLANs. Let I denote the Internet. We treat the Internet as a special VLAN V_I and treat the traffic that a VLAN V_i exchanges with the Internet as the traffic it exchanges with V_I. Let $R_G \in \mathcal{R}$ be the gateway router to the Internet. We treat R_G as the designated router for V_I. Let $\mathcal{X} = (x_{ik})$, $i \in \{1, \ldots, v\} \bigcup \{I\}$, $k \in \{1, \ldots, r\}$ be an indicator variable, whose value is 1 if VLAN V_i chooses router R_k as the designated router, and 0 otherwise. Let $\mathcal{F} = (f_{ij})$, $i, j \in \{1, \ldots, v\} \bigcup \{I\}$, where f_{ij} is the amount of data traffic sent from V_i to V_j. We define $f_{II} = 0$. We note that $\mathcal{F}$ can be derived from the traffic matrix $\mathbf{M_T}$ (Section III-A). Let $\mathcal{D} = (d_{kl})$, $k, l \in \{1, \ldots, r\}$, where d_{kl} is the number of hops in the shortest path between routers R_k and R_l. By definition, $\forall i$, $d_{ii} = 0$. Let $\mathcal{D}^* = (d^*_{ik})$, $i \in \{1, \ldots, v\} \bigcup \{I\}$, $k \in \{1, \ldots, r\}$, where d^*_{ik} is the average number of hops in the shortest paths between a host in V_i and the router R_k. For convenience, we define that $\forall k$, $d^*_{Ik} = 0$. Finally, let $DC(\mathcal{X}, \mathcal{F}, \mathcal{D}, \mathcal{D}^*)$ denote the total amount of data traffic in the network. The router and bridge placement problem is to determine $\mathcal{X}$ such that

$$\text{Minimize} \quad DC(\mathcal{X}, \mathcal{F}, \mathcal{D}, \mathcal{D}^*) + \sum_i \sum_k Br(V_i, R_k) x_{ik} \tag{6}$$

$$\text{subject to:} \quad \forall i, \ \sum_k x_{ik} = 1 \tag{7}$$

$$x_{IG} = 1 \tag{8}$$

$$i, j \in \{1, \ldots, v\} \bigcup \{I\}; \ k, l \in \{1, \ldots, r\}. \tag{9}$$

The second term in (6) represents the total amount of broadcast traffic. Equation (7) is due to the fact that every VLAN must have one and only one designated router and root bridge. Equation (8) is due to the fact that the Internet must always use the gateway router R_G as its designated router.

Data Traffic Cost Model: We next present a model for data traffic costs. In doing so, we separately consider the inter-VLAN traffic of every pair of VLANs and the intra-VLAN traffic of every VLAN

$$\begin{aligned} DC(\mathcal{X}, \mathcal{F}, \mathcal{D}, \mathcal{D}^*) = & \, \mathrm{InterV}(\mathcal{X}, \mathcal{F}, \mathcal{D}, \mathcal{D}^*) \\ & + \mathrm{IntraV}(\mathcal{X}, \mathcal{F}, \mathcal{D}^*). \end{aligned} \tag{10}$$

$\mathrm{InterV}(\mathcal{X}, \mathcal{F}, \mathcal{D}, \mathcal{D}^*)$ and $\mathrm{IntraV}(\mathcal{X}, \mathcal{F}, \mathcal{D}^*)$ are the inter-VLAN and intra-VLAN data traffic, respectively.

- *Inter-VLAN data traffic:* To model the costs associated with inter-VLAN traffic involving VLAN V_i, consider Fig. 4. H_i is a host in VLAN V_i that has designated router R_k. All inter-VLAN traffic sent or received by H_i must traverse the path between H_i and router R_k. In addition, the portion of the traffic exchanged with a given VLAN V_j must traverse the path between R_k and R_l, where R_l is the designated router of VLAN V_j. Finally, the portion of the traffic exchanged with the Internet must traverse the path between R_k and the gateway router to the Internet

$$\begin{aligned} \mathrm{InterV}(\mathcal{X}, \mathcal{F}, \mathcal{D}, \mathcal{D}^*) = & \sum_i \sum_j \sum_k \sum_l f_{ij} d_{kl} x_{ik} x_{jl} \\ & + \sum_i \sum_k \left(d^*_{ik} x_{ik} \sum_{j \neq i} (f_{ij} + f_{ji}) \right). \end{aligned} \tag{11}$$

Intuitively, the first term in (11) represents the inter-VLAN traffic exchanged between the designated routers of VLAN V_i and V_j, summed over all i and j. The second term in (11) represents the total inter-VLAN traffic between a VLAN V_i and all other VLANs, which is exchanged between V_i and its designated router, summed over all i.

- *Intra-VLAN data traffic:* When two hosts in the same VLAN V_i communicate, the number of hops between them depends on the spanning tree of the VLAN and is bounded by two times the total number of hops between each host and the root bridge R_k. In particular, the average number of hops traversed by intra-VLAN traffic is at most two times the average number of hops between a host in V_i and R_k

$$\mathrm{IntraV}(\mathcal{X}, \mathcal{F}, \mathcal{D}^*) = \sum_i \sum_k f_{ii} 2 d^*_{ik} x_{ik}. \tag{12}$$

Finally, using (10)–(12) to rewrite (6), the placement problem may be formulated as follows:

$$\begin{aligned} \text{Minimize} \quad & \sum_i \sum_j \sum_k \sum_l f_{ij} d_{kl} x_{ik} x_{jl} \\ & + \sum_i \sum_k \left(d^*_{ik} x_{ik} \sum_{j \neq i} (f_{ij} + f_{ji}) \right) \\ & + \sum_i \sum_k f_{ii} 2 d^*_{ik} x_{ik} + \sum_i \sum_k Br(V_i, R_k) x_{ik} \end{aligned} \tag{13}$$

$$\text{subject to:} \quad \forall i, \ \sum_k x_{ik} = 1 \tag{14}$$

$$x_{IG} = 1 \tag{15}$$

$$i, j \in \{1, \ldots, v\} \bigcup \{I\}; \ k, l \in \{1, \ldots, r\}. \tag{16}$$

2) Complexity:

Theorem III-2: The router and bridge placement problem is NP-hard with respect to the number of VLANs and the number of routers to choose from.

Proof: The above router and bridge placement problem falls into a category of nonlinear assignment problems, namely quadratic semi-assignment problems (QSAP) [11]. QSAP models the problem of allocating a set of n facilities to a set of m locations, with the costs being the cumulative product of flow between any two facilities and the distance between any two locations, plus the costs associated with a facility being placed at a certain location. The objective is to assign each facility to a location such that the total cost is minimized. QSAP is a variant of the well-known quadratic assignment problem (QAP) [11]. The only difference between QSAP and QAP is that in the former each location may take none, one, or more than one facilities, whereas in QAP each location has to obtain exactly one facility, and vice versa. Both problems are known to be NP-hard [11], [12].

Formally, we are given three matrices with real elements $F = (f_{ij})$, $D = (d_{kl})$, and $B = (b_{ik})$, where f_{ij} is the flow between facility i and facility j, d_{kl} is the distance between location k and location l, and b_{ik} is the cost of placing facility i at location k. Note that F and D matrices can be either symmetric or not. The QSAP can be formulated as follows:

$$\text{Minimize} \quad \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{m}\sum_{l=1}^{m} f_{ij}d_{kl}x_{ik}x_{jl} + \sum_{i=1}^{n}\sum_{k=1}^{m} b_{ik}x_{ik} \tag{17}$$

$$\text{subject to:} \quad \sum_{k=1}^{m} x_{ik} = 1, i = 1, 2, \ldots, n \tag{18}$$

$$x_{ik} \in \{0, 1\}, i = 1, 2, \ldots, n, k = 1, 2, \ldots, m. \tag{19}$$

It is easy to see that our placement problem has the same structure of QSAP. Consider every VLAN as a facility, and every router as a location. Also consider the amount of traffic between VLANs in our problem to be the flow between facilities in QSAP, and the number of hops between routers in our problem to be the distance between locations. Then, the quadratic term of (13), i.e., the term $f_{ij}d_{kl}x_{ik}x_{jl}$, may be viewed as the cost of moving flows between different facilities, i.e., the first term of (17). The rest of the terms of (13) may be viewed as the cost of placing facilities at certain locations, i.e., the second term of (17). Hence, our router and root bridge placement problem can be formulated as QSAP, and thus is NP-hard. ■

3) Heuristic for Router and Bridge Placement: Given the complexity of the problem and the scale of enterprise networks, it is practically impossible for any algorithm to find out the optimum placement. Furthermore, obtaining an accurate estimate of $\mathbf{M_T}$ might be difficult, especially for a network that is yet in operation. We instead design a heuristic that is guided by observations of typical traffic patterns in enterprises. Many enterprises today dedicate a small number of VLANs to house important server machines, such as network file servers and DNS and DHCP servers. These VLANs are likely to be extremely popular in that most hosts in the enterprise communicate with these VLANs. For the vast majority of other nonserver VLANs, however, most traffic exchanged is with the server VLANs and with the Internet. We refer to these nonserver VLANs as client VLANs.

Our solver requires an operator to indicate the set of server VLANs in the design. For every client VLAN, information is provided regarding what fraction of its traffic is exchanged with the Internet and each server VLAN. If this information is unavailable to operators, it is assumed an equal amount of traffic is exchanged with each of the server VLANs.

Consider the terms in (5), (11), and (12). The costs associated with broadcast and intra-VLAN traffic depend entirely on the placement choices of router and root bridge associated with that VLAN alone. The cost associated with inter-VLAN traffic, however, has components that depend on the placement choices of other VLANs. The extent of this dependency on remote VLAN placement is likely higher if there is a strong bias in traffic to the remote VLAN.

The solver proceeds in two steps.

1) Placement decisions are made for all server VLANs. In doing so, terms dependent on placement decisions of other VLANs are not considered.
2) The optimization is conducted for all client VLANs. Given that they primarily communicate with server VLANs, terms involving placement decisions of server VLANs alone are considered, and terms involving placement decisions of other client VLANs are neglected.

With this approach, solving each step above requires minimizing the traffic individually for each VLAN, with the only unknowns being the router and bridge choices for that VLAN. A simple iterative algorithm that tries all possible choices of network elements as designated router and root bridge suffices to ensure the best placement can be found for each VLAN.

IV. Systematic Reachability Control

In this section, we present our approach for systematic reachability control. We first describe the network-wide abstractions that capture the ultimate requirements of reachability control. Next, we formulate the task of ACL placement into a set of optimization problems, each fashioning a different design strategy. We then show that finding the optimal placement is an NP-hard problem. Finally, we present greedy heuristics to approach these optimization problems.

A. Network-Wide Abstractions

We leverage the notion of *Reachability Set* first introduced in 2005 [4]. The reachability set represents the subset of packets (from the universe of all IP packets) that the network will carry from a source to a destination. Formally, we represent a reachability set (RS) for a source u and destination v with a predicate f

$$\mathrm{RS}(u, v) = \{\text{packet } p \mid f(p) = 1\}. \tag{20}$$

For example, the predicate $f(p) = \text{p.src\_addr} \in 10.0/16 \wedge \text{p.dest\_port} \neq 135$ formally defines an RS that contains all packets with source address in the 10.0/16 subnet and a destination port other than 135.

The RS notation has been shown to provide a unifying metric for determining the joint effect of packet filters and routing protocols on end-to-end reachability [4]. It is a natural building block toward a network-wide abstraction that can *completely* capture the operator intent in regard to reachability control. In addition, a network's reachability control policy is said to be resilient against an event if the network continues to uphold the reachability policy despite the occurrence of the event. Putting it together, we model the reachability requirement and the resiliency requirement of a reachability control policy at the granularity of VLANs (or subnets in general) using the following abstractions.

- **Reachability matrix:** Consider a network with N VLANs. The network's reachability policy can be completely described by an N by N reachability matrix, denoted by $\mathbf{M_R}$, where element $\mathbf{M_R}(i,j)$ denotes the maximum RS that will always reach an intended destination host in VLAN j if originated by a host of VLAN i.
- **Managed event set:** The resilience requirement of a network's reachability control policy can be completely described by a managed event set, denoted by $\mathbf{E_m}$, with each element in the set specifying a topology-changing event to which the network must respond without causing the reachability matrix to change.

B. Problem Formulation

The operator's primary task is to place ACLs in a manner that meets the correctness and feasibility criteria.

1) **Correctness criterion:** The network's reachability matrix is invariant and as specified in $\mathbf{M_R}$ under all events in $\mathbf{E_m}$.
2) **Feasibility criterion:**Let $c(r)$ represent the limit on the total number of ACL rules that can be configured on a router r, including all its interfaces and in both traffic directions, without overloading r. Let $b(r)$ be the number of ACL rules that has been configured on router r. Then, $\forall r$, $b(r) \leq c(r)$.

In some network topologies, it may be possible to have multiple ACL placement strategies that meet the correctness and feasibility criteria. For instance, consider a cell of the reachability matrix $\mathbf{M_R}(i,j)$. Consider the simplest case where only a single path of routers exists from VLAN i to VLAN j. The operator may place an ACL permitting only $\mathbf{M_R}(i,j)$ at any of the routers to meet the criteria. We leverage this potential flexibility to allow operators to express their preference for an ACL placement design. In this paper, we consider the following four ACL placement strategies.

- **Minimum rules (MIN) strategy:** The operator wishes to minimize the total number of filter rules installed on all routers in the network. More formally

$$\text{Minimize} \quad \sum_r b(r).$$

- **Load balancing (LB) strategy:** The operator wishes to spread the ACL processing overhead across the network in order to avoid overburdening any router. Formally

$$\text{Minimize} \quad \max_r \{b(r)\}.$$

  The configuration derived from this strategy will not impose a need for costly supernodes. However, the operator may intentionally set $c(r)$ to ∞ when designing a new network (with no hardware purchased yet) or when it is feasible to upgrade existing router hardware.
- **Capability-based (CB) strategy:** The operator wishes to allocate the ACL processing overhead based on each router's filtering capability. Formally

$$\text{Maximize} \quad \min_r \{c(r) - b(r)\}.$$

  Using this strategy, the derived configuration squeezes the most out of the capability of the current hardware.

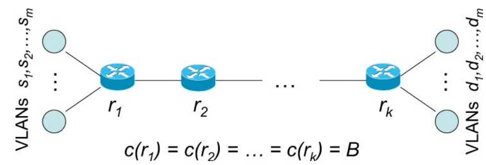

Fig. 5. Network setting used in reduction of the "bin packing" problem.

- **Security centric (SEC) strategy:** The operator wishes to minimize the security risk posed by unauthorized traffic by placing filters as close to the source as possible. For example, early filtering of attack packets aimed at overloading intermediate routers minimizes the number of routers impacted. For a filter f, let $h(f)$ represent the hop count from the router on which f is installed to the gateway router of the traffic sources targeted by f, averaged across all traffic sources. Let H be the average $h(f)$, averaged across all filter rules installed in the network. Ideally, H should be 0. Formally, the goal of the strategy is

$$\text{Minimize} \quad H.$$

C. Complexity of ACL Placement

We model the problem of placing ACLs for the entire reachability matrix $\mathbf{M_R}$ as processing each cell $\mathbf{M_R}(i,j)$ of the matrix one by one until the reachability requirements of all cells are satisfied. The processing for each cell involves finding a correct and feasible placement to install the ACL that effectuates the RS for that cell. Note that if a cell $\mathbf{M_R}(i,j)$ contains "full-reachability" (i.e., any packet from VLAN i can reach VLAN j), the processing for that cell is skipped since no ACL is required. The following theorem establishes that finding the optimal solution to the ACL placement problem is NP-hard.

Theorem IV-1: The ACL placement problem is NP-hard with respect to the number of cells to be processed.

Proof: We present a reduction of the well-known NP-complete "bin packing" decision problem [13] into the problem of ACL placement with the MIN strategy. The reduction holds for the other strategies as well because they share the same decision problem as the MIN strategy.

The "bin packing" decision problem can be formally stated as follows. Given: 1) a finite set U of m items, with each $u \in U$ having a positive integer size $s(u)$; and 2) positive integers B (called the bin capacity) and $k \leq m$, can U be partitioned into k disjoint sets $U_1, \ldots, U_k$ such that for each U_i, the total sum of the sizes of the items in U_i does not exceed B?

Next, we reduce this general problem to the question of whether it is feasible to place ACLs for the special network setting illustrated in Fig. 5. First, we map each of the k bins into a router with $c(r) = B$. The routers form a linear topology that connects two groups of VLANs at the two ends, each with m VLANs. We then map each item $u_i \in U; i = 1, 2, \ldots, m$ to $\mathbf{M_R}(s_i, d_i)$, i.e., one that affects packets originating from VLAN s_i on the left side and going to VLAN d_i on the right, such that the number of ACL rules required for that cell is $s(u_i)$. Finally, we set all the unmapped cells in the reachability matrix to "full reachability," i.e., requiring no packet filter. Clearly, the answer to the "bin packing" problem is yes if and only if it is feasible to place the ACLs for the network setting considered since the ACL for each cell must be placed in one of the routers with sufficient remaining capacity. ■

D. Heuristics for ACL Placement

Since the ACL placement problem is NP-hard, we begin this section by presenting heuristics for processing individual cells (i.e., $\mathbf{M_R}(i,j)$) of the reachability matrix. These fine-grained heuristics provide insights on how our solvers ensure the correctness of placement and approximate various placement strategies. We then discuss placement strategies that involve processing $\mathbf{M_R}$ one row or one column at a time.

1) Placement by Cell: Several polynomial-time heuristics exist for approximating an optimal solution to the "bin-packing" problem. Among them, the "first fit decreasing" strategy, whereby the items are first sorted from largest to smallest and then sequentially placed in the first feasible bin, strikes a good balance between the optimality of the solution and the time complexity. We have adopted the same strategy for ACL placement given a strong resemblance between the two problems. In particular, we first sort the cells in the decreasing order of the number of ACL rules they contain and then process them sequentially using the greedy per-cell placement heuristics presented in the remainder of the section.

To process a given cell $\mathbf{M_R}(i,j)$ of the reachability matrix, we assume that the routing design stage is already completed so that a subgraph $g(i,j)$ of the layer-3 network topology can be derived from the routing design that contains VLANs i and j and satisfies the following conditions.

- The subgraph is sufficiently connected so that no event in $\mathbf{E_m}$ will disconnect VLAN i from VLAN j. That is, we assume that the resilience is ensured by the routing design.
- For each path from VLAN i to VLAN j in the subgraph, either it is one of the default forwarding paths from VLAN i to VLAN j, or there exists an event in $\mathbf{E_m}$ under which it will be used to route traffic from VLAN i to VLAN j.

We note that obtaining $g(i,j)$ may be nontrivial for some of the existing networks where route filters and route redistributions are configured in an *ad hoc* fashion [2]. Here, we assume routing design has been accomplished systematically to ensure the predictability of $g(i,j)$. We also note that overestimating $g(i,j)$—i.e., including more nodes and edges than necessary—does not affect the correctness of the placement, although the resulting solution may place more filter rules than necessary.

The foremost concern of reachability control is the correctness of the solution. The heuristics for all four optimization strategies use the same approach to ensure correctness. They guarantee that the ACL for each cell is placed along all members of an (i,j) edge-cut-set in $g(i,j)$. In other words, all packets that go from VLAN i to VLAN j will encounter an instance of the ACL no matter which physical path they take.

We assume that the address spaces of different VLANs do not overlap and that an algorithm exists to convert $\mathbf{M_R}(i,j)$ into a sequential set $f(i,j)$ of ACL rules. If VLAN i and VLAN j are respectively assigned address blocks of A and B, each rule in $f(i,j)$ looks like the following:

```
{permit or deny} a b [more fields]
```

1: Label all routers with insufficient filter capacity left, i.e., $c(r) - b(r) < n(i,j)$ as ineligible for inclusion into S.
2: Sort R into array based on increasing $b(r)$ values; i.e., $b(R[0]) \le b(R[1]) \le \ldots$; choosing minimum router hop count from i or j as tie breaker
3: $S = \emptyset$;
4: **for** $k = 0$ to $\|R\| - 1$ **do**
5:   Add $R[k]$ to S;
6:   Try finding the smallest edge-cut-set between i and j using only edges connecting a node in S;
7:   **if** successful **then**
8:     {denote the minimum cut-set by CUT}
9:     **for** each edge $e \in CUT$ **do**
10:       **if** both ends of e are routers **then**
11:         **if** starting end of e has smaller $b(r)$ **then**
12:           Add (starting end, 1) to D;
13:         **else**
14:           Add (the other end, 0) to D;
15:         **end if**
16:       **else if** starting end of e is a router **then**
17:         Add (starting end, 1) to D;
18:       **else if** ending end of e is a router **then**
19:         Add (ending end, 0) to D;
20:       **end if**
21:     **end for**
22:     return D;
23:   **end if**
24: **end for**

Fig. 6. ACL placement solver for the LB strategy.

where $a \subseteq A$ and $b \subseteq B$. In addition, to avoid ambiguity, $f(i,j)$ must end with

```
deny A B.
```

Finally, the heuristics require a post-processing step to be performed after the entire reachability matrix is processed. The post-processing step overrides the implicit "`deny any`" on each interface by adding an explicit "`permit any`" at the end of all rules placed on that interface. In addition, the post-processing step may optionally apply compression algorithms [14], [15] to further reduce the number of rules placed on each interface.

Fig. 6 presents the algorithm for the LB strategy. Initially, routers with insufficient capacity to accept $f(i,j)$ are eliminated. The remaining routers are sorted in ascending order of $b(r)$. The number of router hops from either the source or destination VLAN is used as the tie breaker because it is more likely to find small edge-cut-sets closer to the network edge, which is generally less connected than the middle of the topology. The first k routers in the sorted list are considered in set S. The algorithm iterates over k until a minimum edge-cut-set between VLAN i and VLAN j can be found using only edges connecting a node in S. The remaining steps of the algorithm (line 8 onward) identify the appropriate router interfaces on which the filters must be applied. The algorithm can be implemented in polynomial time with well-known efficient polynomial algorithms for finding the minimum edge-cut-set in a network [16].

The heuristics for the other strategies follow the same algorithm with minor variations. The CB strategy simply involves changing the sorting criterion in line 2 from "increasing $b(r)$ values" to "decreasing $(c(r) - b(r))$ values" while keeping the same tie breaker. The SEC strategy involves changing the sorting criterion to "increasing hop count from the gateway router of VLAN i" and changing the tie breaker to "decreasing

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

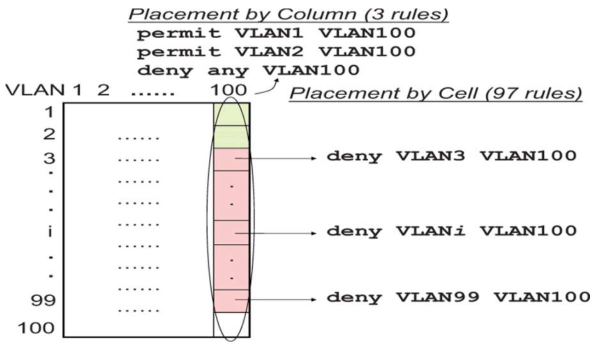

Fig. 7. Hypothetical reachability matrix highlighting the difference between fine-grained and column-based placement.

$(c(r) - b(r))$ values." Finally, the MIN strategy involves replacing lines 2–5 by including all routers in S, and then finding the minimum edge-cut-set.

2) Placement by Row or Column: Our discussion so far assumes a *fine-grained* strategy, where each cell of the reachability matrix is placed independently of other cells. Another degree of freedom for a placement scheme involves placing an entire row or column of the reachability matrix. For instance, security policies such as server access control by nature restrict traffic *to* one VLAN from all other VLANs. For such policies, one strategy is to place the entire column of the reachability matrix corresponding to the destination VLAN. Likewise, security policies like ingress filtering or blocking of unauthorized e-mail servers by nature restrict traffic *from* one VLAN to all other VLANs. In such cases, a potential strategy is to place the entire row of the reachability matrix corresponding to the source VLAN. Note that placement by row/column does not reduce the inherent complexity of finding the optimal solution to the ACL placement problem, which can be shown to remain NP-hard using a similar proof as in Section IV-C.

Placement by row/column offers interesting tradeoffs compared to a fine-grained placement strategy. On the one hand, a fine-grained strategy may distribute rules over multiple routers and require fewer rules on any given router than placement by row/column. In fact, in some scenarios, placement by row/column may not be feasible as the capacity of the router may be exceeded. On the other hand, placement by row/column may offer opportunities to compress the number of rules to be placed by using the wildcard "any" to represent any source or destination. For instance, Fig. 7 shows the reachability matrix for a hypothetical scenario where all hosts in VLANs 1 and 2 have full reachability to VLAN 100 (so no ACL rules are required for the corresponding cells), but all hosts in VLANs 3–99 are denied access to VLAN 100. If cells in the entire column for VLAN 100 are placed together, only three rules are required, as the deny rules from every other source VLAN 3–99 can be effectively compressed using the wildcard "any." However, if a fine-grained strategy is used, potentially 97 rules in total are required to be placed individually, and the rules may be distributed across many routers.

The algorithm in Fig. 6 can be easily extended to process one row or one column of the reachability matrix at a time. The key change is that the target edge-cut-set at line 6 needs to be enlarged to disconnect one source VLAN from many destination VLANs for row-based placement, or one destination VLAN from all source VLANs for column-based placement. Alternatively, the reachability matrix could be processed using a hybrid approach, where some entries are processed by row/column, and others are placed using a fine-grained approach. We omit further details for lack of space.

V. Evaluations and Validation

We evaluate our heuristics on a large-scale campus network with tens of thousands of hosts. The network consists of about 200 routers, 1300 switches, and hundreds of VLANs. Four routers form the core of the network. Typically, each building has a router with a link to one of the core routers. This link connects all hosts in the building to the rest of the network. Our data includes configuration files of all switches and routers and the physical topology of the network.

VLAN Usage: While the campus IT operators provide routing services for the entire campus, each logical group such as the School of Engineering, the School of Liberal Arts, and the Libraries has its own administrators. Each administrative unit is given an IP address block and is free to assign addresses within that block to individual hosts. The operator policy requires that hosts in different administrative units must belong to different VLANs. VLANs are extensively used to meet this goal, as well as to constrain the size of broadcast domains. Most VLANs span a small section of the campus—about 50% of them span only one building. However, about 10% of the VLANs span 5+ buildings, and the largest VLAN spans over 60 buildings. VLANs with a large span correspond to administrative units that have hosts in most buildings on campus, e.g., hosts in all classrooms are administered together and are grouped into a VLAN.

ACL Usage: Prominent ACL policies used by the campus network include: 1) ingress filtering to ensure that packets have a source IP address from the address space of their originating subnets; 2) restricting communication involving dormitory hosts; 3) restrictions involving wireless traffic; and 4) restricting communication with data centers that house many key servers. Overall, ACL rules are placed in over 70 routers, with about 20% of the routers having 300+ rules, which may include rules from multiple ACLs.

A. VLAN Design

In this section, we present results evaluating our systematic design approach for each of the VLAN design tasks.

Grouping Hosts Into VLANs: With help from the operators, we categorize the hosts on a large segment of the campus. Each category corresponds to a different administrative unit. In total, there are 119 categories and 15 084 hosts. Many categories are small, and the median category has only 79 hosts. However, the largest category includes 2000+ hosts.

We group hosts into VLANs using our systematic approach. Our algorithms are subject to two constraints. First, a maximum of 182 VLANs is permitted, as this is the number of VLANs used in the current design. Second, hosts from different categories are required to belong to different VLANs.

Table II shows the number of hosts per VLAN produced by our approach and compares the results to the current design. The results show the effectiveness of our approach in avoiding the creation of large VLANs with many hosts. The maximum number of hosts in any VLAN is reduced from 254 to 195, and the 90th percentile is reduced from 193 to 167. This is achieved

TABLE II
NUMBER OF HOSTS PER VLAN WITH THE CURRENT AND THE SYSTEMATIC DESIGNS

	# Hosts per VLAN (182 VLANs)	
	Current	Systematic
Mean	82.9	82.9
Std Dev	71.9	57.1
90%ile	193	167
Max	254	195

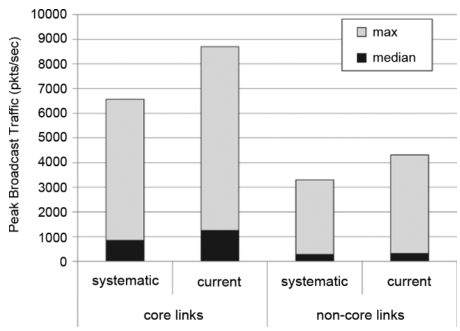

Fig. 8. Estimated peak broadcast traffic load per link.

by a more equitable distribution of hosts across VLANs as indicated by the lower standard deviation. In addition, we also found (though not shown in the table) that our systematic approach also reduces the span of large VLANs by decreasing the number of links in their spanning trees. In particular, the maximum number of spanning tree links in any VLAN is reduced from 417 to 254.

We next study the potential benefit of our systematic grouping in reducing broadcast traffic, which is usually dominated by VLANs with a large size and span. To get a realistic estimate of broadcast traffic pattern, we measured the broadcast traffic sent by hosts in one of the VLANs over a 24-h period. We observed an average and peak packet rate of 0.004 and 2.12 packets/s/source, respectively. We then estimated the peak broadcast traffic seen per link, assuming every host generates broadcast traffic at the peak rate.

Fig. 8 shows the median and maximum estimated peak broadcast packet rates per network link for the current grouping and our systematic grouping. Two types of links, *core links* and *noncore links*, are shown. The core links include links between core routers and links connecting a core router to routers of various buildings in campus. All the remaining links are noncore links. Overall, there are about 500 core links and 41 000 noncore links. Our systematic design results in similar median broadcast traffic to the current design, but significantly reduces the maximum broadcast traffic rate by around 1000 and 2000 packets/s for noncore links and core links, respectively. The decrease of broadcast traffic in core links comes from both reducing the number of hosts in large VLANs as well as ensuring VLANs span as few links as possible. The drop in broadcast packet rate on core links allows core routers to potentially save their processing power for more important tasks, e.g., assuring critical traffic is quickly transported through the backbone.

Router and Bridge Placement: The operators provided a set of six server VLANs that housed many popular servers that other hosts would access. These include campus Web servers, DNS and DHCP servers, and other important data servers. The operators also confirmed that a large portion of traffic from the other VLANs (client VLANs) is either exchanged with these server VLANs or with the Internet. We then compute the optimal placement of their routers using our algorithm in Section III-C. We assume that intra-VLAN data traffic is negligible, and 1% of inter-VLAN data traffic incurs broadcast traffic. Among the remaining 99% of inter-VLAN data traffic, f% is exchanged with the Internet, and the rest is exchanged evenly with each server VLAN. We believe these models are realistic in many enterprise settings, and the operators confirmed these are reasonable traffic models.

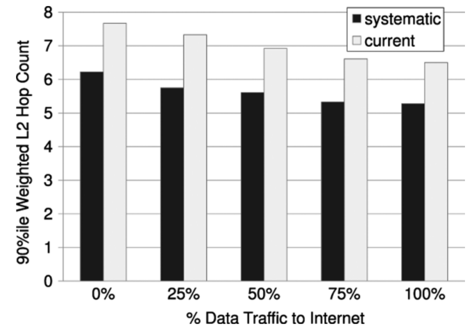

Fig. 9. Reduction of hops traversed by data traffic using our systematic router placement, with varying f (i.e., fraction of traffic to the Internet).

Fig. 9 explores the effectiveness of our systematic router placement in reducing the number of hops traversed by data traffic when f is varied. There are two bars for each choice of f, one for the current placement and the other for our systematic placement. Each bar represents the 90th percentile of the average weighted hop count for hosts in a client VLAN. The weighted hop count is the average number of hops from a client host to the gateway routers of the server VLANs and the Internet, weighted by the corresponding fraction of data traffic exchanged with them. For all scenarios, the average weighted hop count is decreased by 1–1.5 hops using our systematic placement since our systematic approach takes traffic patterns into account. Reducing the number of hops traversed by data traffic not only results in lower delays, but also reduces the possibility of communication being disrupted by failures. Furthermore, the data traffic carried by network links could also be reduced.

We next study the potential benefit of our systematic placement in reducing data traffic on network links. To model the traffic behavior of end-hosts, we consider two models: a uniform model and a trace model. The uniform model assumes every host transmits data uniformly at 10 kb/s. The trace model is based on traffic traces collected at Lawrence Berkeley National Laboratory (LBNL), Berkeley, CA [17]. The traces were recorded over a 22-h period in December 2004, covering about 8000 internal addresses. We computed a list of average data rate sent/received by each internal address, which ranges from 0–8183 kb/s with a mean of 14.6 kb/s. We then randomly assigned a rate from this list to each host in our campus network and evaluated the traffic load on each link. Fig. 10 shows the median and 95th percentile traffic load on the core links using both traffic models under the current and systematic designs. While the median core link load is similar for both designs using the two traffic models, our systematic placement improves the 95th percentile load from 20.9 to 6.4 Mb/s and from 27 to

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

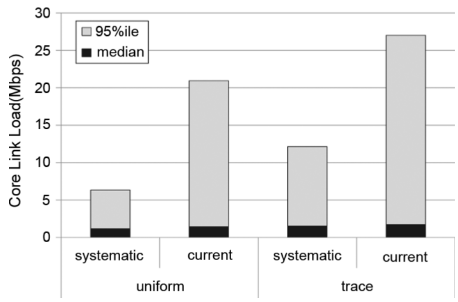

Fig. 10. Data traffic load on core links using the uniform and the trace traffic models, with $f = 50$.

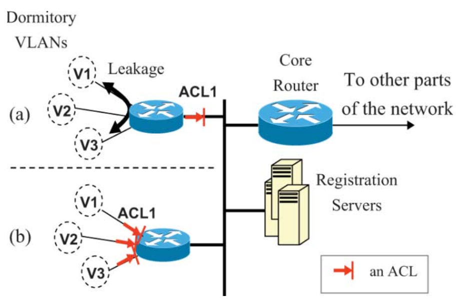

Fig. 11. (a) Scenario of ACL placement inconsistent with intent. (b) Corrected placement.

12.1 Mb/s for the uniform model and the trace model, respectively. The results show that shorter data paths may involve traversal of fewer core links, and the potential reductions in data traffic on these core links is significant.

B. Placement of ACL Rules

The campus network we analyzed is well run, and many hours of design time have been spent on its ACL rules. Using our systematic design algorithms, we were able to automatically create an ACL placement that mostly matches the current placements in this large-scale network using only an hour of CPU time. Beyond the general time savings in creating placements and adapting them as the network changes, we found two interesting examples that illustrate the importance and benefits of systematic placement of ACL rules.

Correctness of Placement: Our analysis discovered an inconsistency between operator intent and the current ACL placement. One operator policy is to prevent access from unregistered dormitory users to any host other than a small number of well-known registration servers. Fig. 11(a) illustrates the relevant segment of the network. Hosts in the dormitories are separated into a group of VLANs. These VLANs share the same gateway router. The gateway router and a core router are part of a broadcast subnet. In order to regulate the traffic, the operators applied an ACL on the outbound interface from each router to the broadcast subnet. However, this decision results in leakage of undesirable traffic from unregistered users in one VLAN to other VLANs that share the same designated router. Since some routers are the first-hop gateways for over 20 VLANs, undesired communication is being permitted between a large number hosts. The operators confirmed that systematic design had identified a previously unknown error in their ACL placement and thanked us for pointing it out.

Fig. 11(b) illustrates a correct placement. It involves duplicating and moving the ACL to each inbound VLAN interface and could result in significantly more rules. We hypothesize that the inconsistency arose as the operators tried to cut the number of rules in an *ad hoc* fashion. Such errors can be easily avoided by systematic design approaches.

TABLE III
PLACEMENT OF ACL RULES BASED ON VARIOUS OPERATOR OBJECTIVES UNDER TWO EXTREME RESOURCE CONSTRAINTS

Metrics $b(r)$=# rules on r $c(r)$=ACL capacity of r	$c(r) = \infty$					$c(r) \leq 300$				
	By Col.	Fine-Grained MIN	Fine-Grained LB	Fine-Grained CB	Fine-Grained SEC	By Col.	Fine-Grained MIN	Fine-Grained LB	Fine-Grained CB	Fine-Grained SEC
$\sum_r b(r)$	693	**1169**	2434	1169	1169	N/A	**1369**	2408	2389	1369
$\max_r\{b(r)\}$	693	418	**280**	1169	418	N/A	280	**280**	280	280
$\min_r\{c(r) - b(r)\}$	∞	∞	∞	∞	∞	N/A	20	20	**20**	20
H	1.69	0	0.1	2.06	**0**	N/A	0	0.09	0.08	**0**

Customizing Placement for Operator Objectives: To illustrate our systematic approach for customizing ACL placement, we consider the largest ACL in the campus network. This ACL consists of 693 rules The ACL policy permits a specified list of hosts across various client VLANs to access a server VLAN; all other hosts are denied access to the server VLAN.

In the current design, all rules are placed in the last-hop router to the destination server VLAN. While this is a reasonable placement, there are alternative strategies that may be of interest to an operator. For instance, an operator may prefer to drop unwanted traffic closer to the source or may wish to reduce the total rules placed on the router.

Table III illustrates how our approach can enable an operator to flexibly choose from a range of placement strategies based on the desired criteria of interest. Each column corresponds to a placement scheme, and each row corresponds to the metric used to rate a placement scheme.

The left half of the table presents results with these schemes assuming no constraints on the number of rules that may be placed on any router $(c(r) = \infty)$. One of our strategies (column-based placement) does match the design currently employed in the network. This strategy performs best in terms of keeping the total rules across the network small, for reasons elaborated in Section IV-D.2. However, other strategies offer benefits in alternate metrics of interest to the operator. For instance, the fine-grained SEC strategy pushes all rules to the first-hop router $(H = 0)$, ensuring that traffic is filtered as early as possible, while the LB strategy ensures the maximum number of rules in any router is at most 280.

In networks built with low-end routers, it may not be feasible to place all rules in one router. To show the potential value of our systematic approach in such environments, we limit the processing capability of all routers in the network to be fewer than 300 rules $(c(r) \leq 300)$. The right half of Table III presents the results from systematic placement in this regime. Unlike column-based placement, all fine-grained strategies are able to produce a feasible placement despite the tight constraint. In addition, the various strategies offer benefits in metrics they target. For instance, the MIN strategy ensures the total number of rules is small (1369). Interestingly, the strategy also performs well in the other metrics.

Fig. 12 depicts how rules are distributed in the network after applying the fine-grained LB strategy in this setting. Only routers and relevant VLANs (i.e., the server VLAN, and client VLANs with permitted hosts to the server VLAN) are shown. The number of rules varies per router, depending on the topology and the number of client VLANs attached to

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

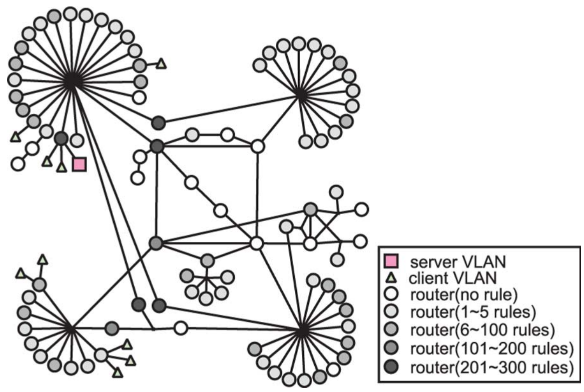

Fig. 12. Layer-3 topology showing systematic distribution of ACL rules after applying fine-gained, LB placement strategy.

the router. Overall, the LB strategy spreads the load across the network, with no router having more than 280 rules. This exhibits the potential to systematically design the placement for the entire network with only lower-end hardware.

VI. Related Work

Many prior efforts on systematic network design focus on tasks encountered in carrier networks, such as configuring BGP policies [7]–[9], [18], optimizing OSPF weights, and redundancy planning [19]. In contrast, we focus on tasks in enterprise networks, which has received limited attention.

A few recent studies [20]–[25] are partially motivated by enterprise networks. Most of them consider clean-slate designs by rearchitecting the control plane itself to contain the complexity of network design. In contrast, our work is relevant to both existing enterprise environments and clean-slate designs.

Industry-driven efforts to simplifying enterprise network configuration involve template-based approaches [26]–[30] and abstract languages to specify configurations in a vendor-neutral fashion [31]–[33]. However, these approaches merely model the low-level mechanism and configuration and do not abstract high-level operator intent.

A logic-based approach to configuration generation based on model-finding is presented in [6]. The focus is on the generation of correct configurations, and the system does not support optimization to meet desired performance objectives. Many works have approached the problem of minimizing rules in a single ACL (e.g., [14]). In constrast, we focus on distributing ACL rules in a network to realize a given reachability policy. Previous work including our own has looked at bottom-up configuration analysis in the context of VLAN design [3] and network reachability policies [4], [34]. In contrast, our focus in this paper is on systematic design in these areas.

VII. Conclusion

In this paper, we have shown the viability and importance of a systematic approach to two key design tasks in enterprise networks: VLAN design and reachability control. Our contributions include: 1) a systematic formulation of these critical but poorly understood enterprise design tasks; 2) a set of algorithms to solve the formulated problems; and 3) a validation of the systematic approach on a unique large-scale campus network data set.

Our evaluations show the promise of our approach. The campus network we analyzed is well run, and many hours of human design time have been spent on it. Yet, our approach produces better results with less human effort. Beyond the general time savings in the design process, a systematic approach can ensure correctness and lead to significantly better designs. For example, through systematic VLAN design, broadcast and data traffic on the core links of the campus network can be reduced by over 24% and 55%, respectively. Systematic placement of ACLs ensures the design correctly conforms to the operator's security objectives. In contrast, today's *ad hoc* design processes can result in inconsistencies such as those we pointed in our analysis. Finally, our approach can be customized to optimize for operator-preferred design strategies and can produce designs tailored to network parameters such as traffic patterns and router resource constraints.

While this paper has focused on greenfield networks, i.e., new networks to be deployed for the first time, our approach also lends itself to dealing with the evolution of the network after the initial systematic deployment [35]. In the future, we hope to gain experience with our approach on a wider range of enterprise networks and apply the systematic approach to other enterprise tasks such as routing design. One open question is whether there exists a general method for finding suitable network-wide abstractions to model the operational goals of different networks and design tasks. Another open question is how to best integrate the solutions for different tasks whose design space may overlap. For example, a particular choice of routing design may impact how optimal a solution our packet filter placement heuristics can achieve.

We view our work as an important component of an overall system to automatically translate operator intent to low-level configuration files. A complete system must include: 1) ways to translate operational goals into network-wide abstractions (e.g., through GUIs); 2) mechanisms to obtain baseline data such as the traffic matrix, and reachability matrix, either through measurements or static analysis of existing network configurations [4]; and 3) tools similar to PRESTO [30] to convert systematic design solutions into device-vendor-specific configuration commands. We defer further investigation of such a comprehensive solution to future work.

Acknowledgment

The authors thank B. Devine, D. Kyburz, and other colleagues in the Information Technology Department at Purdue (ITaP), West Lafayette, IN, for providing access to the data and for being generous with their time. They also thank D. Collins and S. Dath Krothapalli for their help in evaluating their systematic design.

References

[1] 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*, 2004, pp. 27–40.

[2] 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. ACM SIGCOMM*, 2008, pp. 39–50.

[3] P. Garimella, Y.-W. E. Sung, N. Zhang, and S. Rao, "Characterizing VLAN usage in an operational network," in *Proc. ACM SIGCOMM INM*, 2007, pp. 305–306.

[4] G. Xie, J. Zhan, D. A. Maltz, H. Zhang, A. Greenberg, G. Hjalmtysson, and J. Rexford, "On static reachability analysis of IP networks," in *Proc. IEEE INFOCOM*, 2005, vol. 3, pp. 2170–2183.

[5] Z. Kerravala, "Configuration management delivers business resiliency," The Yankee Group, Boston, MA, 2002.
[6] S. Narain, "Network configuration management via model finding," in *Proc. LISA*, 2005, vol. 19, p. 15.
[7] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessensand, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra, "Routing Policy Specification Language (RPSL)," Internet Engineering Task Force, RFC 2622, 1999.
[8] T. G. Griffin and J. L. Sobrinho, "Metarouting," in *Proc. ACM SIGCOMM*, 2005, pp. 1–12.
[9] J. Gottlieb, A. Greenberg, J. Rexford, and J. Wang, "Automated provisioning of BGP customers," *IEEE Network*, vol. 17, no. 6, pp. 44–55, Dec. 2003.
[10] M. R. Garey and D. S. Johnson*, Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: Freeman, 1979.
[11] R. E. Burkard, E. Çela, P. M. Pardalos, and L. S. Pitsoulis, "Quadratic assignment problems," *Eur. J. Oper. Res.*, vol. 15, no. 3, pp. 283–289, Mar. 1984.
[12] S. Voss, "Heuristics for nonlinear assignment problems," in *Nonlinear Assignment Problems: Algorithms and Applications*. Dordrecht, The Netherlands: Kluwer, 2000, pp. 175–215.
[13] P. E. Dunne, "An annotated list of selected NP-complete problems," 2008 [Online]. Available: http://www.csc.liv.ac.uk/~ped/teachadmin/COMP202/annotated_np.html
[14] A. Liu, E. Torng, and C. Meiners, "Firewall compressor: An algorithm for minimizing firewall policies," in *Proc. IEEE INFOCOM*, 2008, pp. 176–180.
[15] S. Acharya, J. Wang, Z. Ge, T. Znati, and A. Greenberg, "Simulation study of firewalls to aid improved performance," in *Proc. ANSS*, Washington, DC, 2006, pp. 18–26.
[16] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson*, Introduction to Algorithms*. New York: McGraw-Hill, 2001.
[17] R. Pang, M. Allman, M. Bennett, J. Lee, V. Paxson, and B. Tierney, "A first look at modern enterprise traffic," in *Proc. ACM SIGCOMM IMC*, 2005, p. 2.
[18] H. Boehm, A. Feldmann, O. Maennel, C. Reiser, and R. Volk, "Network-wide inter-domain routing policies: Design and realization," 2005.
[19] R. Rastogi, Y. Breitbart, M. Garofalakis, and A. Kumar, "Optimal configuration of OSPF aggregates," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 181–194, Apr. 2003.
[20] M. Caesar, D. Caldwell, N. Feamster, J. Rexford, A. Shaikh, and J. van der Merwe, "Design and implementation of a routing control platform," in *Proc. NSDI*, 2005, vol. 2, pp. 15–28.
[21] N. Feamster, H. Balakrishnan, J. Rexford, A. Shaikh, and J. van der Merwe, "The case for separating routing from routers," in *Proc. ACM SIGCOMM Workshop Future Directions Netw. Arch.*, 2004, pp. 5–12.
[22] M. Casado, T. Garfinkel, A. Akella, M. Freedman, D. Boneh, N. McKeown, and S. Shenker, "SANE: A protection architecture for enterprise networks," in *Proc. USENIX Security*, 2006, Article no. 10.
[23] A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, J. Rexford, G. Xie, H. Yan, J. Zhan, and H. Zhang, "A clean slate 4D approach to network control and management," *Comput. Commun. Rev.*, vol. 35, no. 5, pp. 41–54, Oct. 2005.
[24] J. Rexford, A. Greenberg, G. Hjalmtysson, D. A. Maltz, A. Myers, G. Xie, J. Zhan, and H. Zhang, "Network-wide decision making: Toward a wafer-thin control plane," in *Proc. ACM SIGCOMM HotNets*, Nov. 2004.
[25] H. Ballani and P. Francis, "Conman: A step towards network manageability," in *Proc. ACM SIGCOMM*, 2007, pp. 205–216.
[26] "Cisco IP solution center," Cisco, San Jose, CA [Online]. Available: http://www.cisco.com/en/US/products/sw/netmgtsw/ps4748/index.html
[27] "Intelliden," IBM, Armonk, NY [Online]. Available: http://www.intelliden.com/
[28] "Opsware," Hewlett-Packard Company, Palo Alto, CA [Online]. Available: http://www.opsware.com/
[29] "Voyence," EMC, Hopkinton, MA [Online]. Available: http://www.voyence.com/
[30] 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. USENIX*, 2007, Article no. 6.
[31] J. Case, M. Fedor, M. Schoffstall, and J. Davin, "A simple network management rrotocol (SNMP)," RFC 1157, 1990.
[32] "Distributed Management Task Force, Inc.," Portland, OR [Online]. Available: http://www.dmtf.org
[33] "CPE WAN management protocol," DSL Forum, TR-069, 2004 [Online]. Available: http://www.broadband-forum.org/technical/download/TR-069.pdf
[34] A. R. Khakpour and A. X. Liu, "Quantifying and querying network reachability," in *Proc. ICDCS*, 2010, pp. 817–826.
[35] X. Sun, Y.-W. E. Sung, S. Krothapalli, and S. Rao, "A systematic approach for evolving VLAN designs," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

Yu-Wei Eric Sung was born in Taipei, Taiwan, in 1980. He received the B.A.Sc. degree in computer engineering from the University of Toronto, Toronto, ON, Canada, in 2004 and the M.S. and Ph.D. degrees in electrical and computer engineering from Purdue University, West Lafayette, IN, in 2006 and 2010, respectively.

He is a Software Engineer with the Net Systems Team of Facebook, Palo Alto, CA, where he helps build the infrastructure, systems, and tools that aid in designing, optimizing, and maintaining Facebook's network from data center to end-user. His research interests include peer-to-peer systems, enterprise network management, and cloud computing.

Xin Sun received the B.Eng. degree in computer engineering from University of Science and Technology of China, Hefei, China, in 2005 and is currently a Ph.D. candidate with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, where he works with Prof. Sanjay Rao.

His research interests are in the design and configuration of large-scale enterprise networks and in the migration of such networks to new architectures. In the past, he has also worked on security issues of peer-to-peer systems.

Mr. Sun is a recipient of a Best Bachelor's Thesis Award from University of Science and Technology of China.

Sanjay G. Rao received the Bachelor's degree in computer science and engineering from the Indian Institute of Technology, Madras, India, in 1997 and the Ph.D. degree from the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, in 2004.

He is an Assistant Professor with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, where he leads the Internet Systems Laboratory. He was a Visiting Researcher with the Network Measurement and Management Group, AT&T Research, Florham Park, NJ, during summer 2006. He has played a leadership role in the End System Multicast project that pioneers P2P live-streaming, which is now a mainstream research area and an emerging commercial sector. His research interests are in networking, more specifically in enterprise network management, cloud computing, and peer-to-peer systems.

Prof. Rao has served on the Technical Program Committees of several workshops and conferences, including ACM SIGCOMM, IEEE INFOCOM, and ACM CoNEXT, and was the Technical Program Co-Chair of the INM/WREN workshop on Internet network management and enterprise networks held in conjunction with NSDI 2010. He is a recipient of the NSF CAREER Award.

Geoffrey G. Xie received the B.S. degree in computer science from Fudan University, Shanghai, China, in 1986, the M.S. degree in computer science and the M.A. degree in mathematics from Bowling Green State University, Bowling Green, OH, in 1988, and the Ph.D. degree in computer sciences from the University of Texas, Austin, in 1996.

He is a Professor with the Computer Science Department, U.S. Naval Postgraduate School, Monterey, CA. He was a Visiting Scientist with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, from 2003 to 2004. He has published over 60 articles in various areas of networking. His current research interests include clean-slate design of IP control plane, static analysis of network configuration, routing theories, underwater acoustic networks, and abstraction-driven design and analysis of enterprise networks.

Prof. Xie was an Editor of *Computer Networks* from 2001 to 2004. He co-chaired the ACM SIGCOMM Internet Network Management Workshop in 2007 and is currently a member of the workshop's Steering Committee.

David A. Maltz received the S.B. and S.M. degrees in electrical engineering and computer science from Massachusetts Institute of Technology (MIT), Cambridge, MA, in 1993 and 1994, respectively, and the Ph.D. degree in computer science from Carnegie Mellon University (CMU), Pittsburgh, PA, in 2001.

He is with the Networking Research Group of Microsoft Research, Seattle, WA, where he designs new architectures for data center and enterprise networks and investigates techniques for managing IT infrastructure. He founded a traffic engineering startup and was a Post-Doctoral Fellow with CMU.