Management Plane Analytics

*Aaron Gember-Jacobson, *Wenfei Wu, *Xiujun Li, *Aditya Akella, †Ratul Mahajan

*University of Wisconsin-Madison, †Microsoft Research

ABSTRACT

While it is generally held that network management is tedious and error-prone, it is not well understood which specific management practices increase the risk of failures. Indeed, our survey of 51 network operators reveals a significant diversity of opinions, and our characterization of the management practices in the 850+ networks of a large online service provider shows significant diversity in prevalent practices. Motivated by these observations, we develop a management plane analytics (MPA) framework that an organization can use to: (i) infer which management practices impact network health, and (ii) develop a predictive model of health, based on observed practices, to improve network management. We overcome the challenges of sparse and skewed data by aggregating data from many networks, reducing data dimensionality, and oversampling minority cases. Our learned models predict network health with an accuracy of 76-89%, and our causal analysis uncovers some high impact practices that operators thought had a low impact on network health. Our tool is publicly available, so organizations can analyze their own management practices.

Categories and Subject Descriptors

C.2.3 [**Computer-Communication Networks**]: Network Operations—*Network management*

Keywords

Network management practices; network health; quasi-experimental design; decision trees

1. INTRODUCTION

Computer networks are logically composed of three planes: data, control, and management (Figure 1). The data plane forwards packets. The control plane generates forwarding tables and filters for the data plane using configuration files and routing protocols (e.g., OSPF and BGP)—or control programs in the case of software defined networking (SDN). The management plane is a collection of practices that define the network's physical composition, control plane configuration, and monitoring schemes based on an organization's policies and objectives.

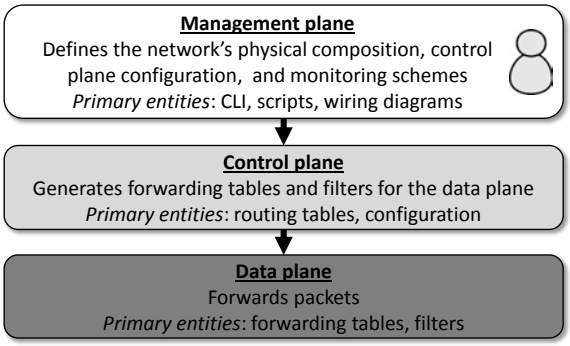


Figure 1: The three network planes

The networking community has a strong track record of developing innovative tools and techniques to discover how control and data planes function, even when the network does not directly reveal that information. Examples include techniques to infer the paths taken by packets [35], link characteristics [10], available bandwidth along a path [16], loss rate and re-ordering [24], network topology [31], and so on.

However, little work has gone into characterizing the management plane, despite its importance to well-functioning networks. As a result, we lack a systematic understanding of what *management practices* are common today—i.e., how do operators design and (re)configure the physical and logical structure of their networks? Even simple practices such as how heterogeneous are a network's devices, and how often and why are networks changed, are poorly understood in the research community. Furthermore, while it is generally held that network management is tedious and error-prone, researchers and operators alike don't have a principled understanding of which management practices pose a higher risk of performance and availability problems. Indeed, our survey of 51 network operators (Figure 2) reveals a significant diversity of opinions regarding this issue.

This paper makes two contributions. First, we present a systematic characterization of the management practices employed in over 850 networks managed by a large online service provider. Our characterization offers the first in-depth look into the management practices used in modern networks. Second, we propose a *management plane analytics* (MPA) framework that uncovers the relationship between network health (e.g., the frequency of performance and availability problems) and management practices. An organization can apply MPA to its networks to: (i) determine which practices *cause* a decline, or improvement, in network health; and

IMC'15, October 28–30, 2015, Tokyo, Japan.

DOI: http://dx.doi.org/10.1145/2815675.2815684.

(ii) develop a predictive model of health, based on management practices, to help shape future practices and aid what-if analysis.

In our work we face three main challenges. First, management practices and their impact are rarely directly logged. We show how management practices and network health can be *inferred* from other data, including inventory records, snapshots of device configurations, and trouble ticket logs. This data is indirect and noisy, but useful information can be extracted from it.

The second challenge is dealing with a limited amount of data. For any given network in an organization (e.g., a data center network hosting a certain Web service) the number of available snapshots of its design and operation may be limited, and some snapshots may be missing due to incomplete or inconsistent logging. Our key insight is to identify causal relationships and build predictive models by aggregating data from many networks and many months. This aggregation eliminates noise from individual networks and provides a broader picture of an organizations' practices.

Finally, we show that skew in network operations data makes it challenging to identify causal relationships and build accurate predictive models. Common approaches for decomposing dependencies (e.g., ANOVA [3] or ICA [9]) are unsuitable, because relationships between network health and management practices may be non-monotonic. Similarly, quasi-experimental designs (QEDs) that rely on exact matching to eliminate the effects of confounding practices [21] do not work, because many management practices are strongly related. Furthermore, predictive models constructed using standard learning algorithms (e.g., decisions trees built using C4.5 [27]) inaccurately predict situations that lead to poor network health due to the presence of substantially more "healthy network" cases in the data.

We use three main techniques to overcome this skew. First, we use *mutual information* to uncover statistical dependencies between network health and management practices. Mutual information quantifies the extent to which knowing a management practice reduces uncertainty about network health. Second, we use QEDs based on *propensity score matching* [33] to discover causal relationships between management practices and network health. Propensity scores systematically account for the bias in treatment selection caused by confounding practices. Finally, when learning predictive models, we use oversampling and boosting [12] to improve accuracy for the minority (i.e., "unhealthy network") cases.

Key findings from applying our methods to several hundred networks of a large online service provider (OSP) are:

- There is significant variation in management practices across the networks, even though they are all managed based on the same set of recommended guidelines. For instance, we find networks differ substantially with respect to hardware and firmware heterogeneity, the extent of automation used, and the way changes are made to them.
- By applying MPA to the OSP's data, we determine that number of devices, number of change events, and number of change types have a strong statistical dependence and causal relationship with network health (quantified using number of tickets). While in some cases our causal analysis agrees with operator feedback (e.g., number of change events has high impact), in other cases it contradicts them (e.g., the fraction of changes where an ACL is modified has moderately high impact, despite a majority opinion that its impact is low).
- Decision trees for predicting two coarse-grained health classes have 91% (cross-validation) accuracy, whereas those for five fine-grained classes have 81% accuracy. While our enhancements improve multi-class accuracy significantly, fine-grained predictions are still suboptimal due to a lack of sufficient data. When applied in an online fashion, our 2-class (5-class) model can predict network health with 89% (76%) accuracy.

Our work is a step toward designing a better management plane that reduces the burden on operators and reduces the frequency of failures. Although the observations we make for the OSP's networks may not apply to all networks—due to differences between organizations and types of networks (e.g., data centers vs. wide area networks)—our publicly available MPA framework [2] can be applied to any set of networks to identify the extent to which particular management practices impact the health of those networks. Now is a particularly relevant time for this undertaking because, in the form of SDN, the community is engaged in re-architecting networks. By providing operators a detailed understanding of the strengths and weaknesses of their current management plane, our work can help inform the design of the next generation management plane.

2. INFERRING MANAGEMENT PRACTICES

Our goal is to design a framework that an organization that operates a collection of networks (e.g., a university campus, or an online service provider) can use to understand and improve its management plane. Organizations that manage a large number of devices typically do not view all devices as belonging to one network, but instead view the devices as partitioned across multiple networks. A *network* in this context is a collection of devices that either connects compute equipment that hosts specific workloads or connects other networks to each other or the external world. A *workload* is a service (e.g., a file system, or an application) or a group of users (e.g., students using PCs in a department).

A challenge in designing our desired framework is that management practices are not explicitly logged. While the control and data planes can be queried to quantify their behavior [10, 16, 24, 31, 35], no such capability exists for the management plane. This gap stems from humans being the primary actors in the management plane. Operators translate high-level intents into a suitable setup of devices, protocols, and configurations to create a functional, healthy network. Even when recommended procedures are documented, there is no guarantee that operators adhere to these practices.

Fortunately, we are able to infer management practices from other readily available data sources. In this section, we describe these sources and the management practice *metrics* we can infer.

2.1 Data Sources

We can infer management practices and network health from three data sources that are commonly available. Such data sources have already been used in prior work, albeit to study a limited set of management practices [6, 20, 26, 34]. We build upon these efforts to provide a more thorough view of management practices and their relationship to network health. The data sources are:

1) Inventory records. Most organizations directly track the set of networks they manage, and the role the networks play. They also record the vendor, model, location, and role (switch, router, load balancer, etc.) of every device in their deployment, and the network it belongs to. This data can be used to infer a network's basic composition and purpose.

2) Device configuration snapshots. Network management systems (NMS) track changes in device configurations to aid network operators in a variety tasks, such as debugging configuration errors or rolling back changes when problems emerge. NMSes such as RANCID [28] and HPNA [36] subscribe to syslog feeds from network devices and snapshot a device's configuration whenever the

Design practices
$D1$. Number of services, users, or networks connected
$D2$. Number of devices, vendors, models, roles (e.g., switch, router, firewall), and firmware versions
$D3$. Hardware and firmware heterogeneity
$D4$. Number of data plane constructs used (e.g., VLAN spanning tree, link aggregation), and instance counts
$D5$. Number and size of BGP & OSPF routing instances
$D6$. Intra- and inter-device config reference counts
Operational practices
$O1$. Number of config changes and devices changed
$O2$. Number of automated changes
$O3$. Number and modality of changes of specific types (e.g., interface, ACL, router, VLAN)
$O4$. Number of devices changed together

Table 1: Management practice metrics

device generates a syslog alert that its configuration has changed. Each snapshot includes the configuration text, as well as metadata about the change, e.g., when it occurred and the login information of the entity (i.e., user or script) that made the change. The snapshots are archived in a database or version control system.

3) Trouble ticket logs. When users report network problems, or monitoring systems raise alarms, a trouble ticket is created in an incident management system. The ticket is used to track the duration, symptoms, and diagnosis of the problem. Each ticket has a mix of structured and unstructured information. The former includes the time the problem was discovered and resolved, the name(s) of device(s) causing or effected by the problem, and symptoms or resolutions selected from pre-defined lists; the latter includes syslog details and communication (e.g., emails and IMs) between operators that occurred to diagnose the issue.

2.2 Metrics

Using these data sources, we can infer management practices and network health, and model them using *metrics*. We broadly classify management practices into two classes (Table 1): *design practices* are long-term decisions concerning the network's structure and provisioning (e.g., selecting how many switches and from which vendors); *operational practices* are day-to-day activities that change the network in response to emerging needs (e.g., adding subnets).

Design Practices. Design practices influence four sets of network artifacts: the network's purpose, its physical composition, and the logical structure and composition of its data and control planes. The metrics we use to quantitatively describe a network's purpose and its physical composition are rather straightforward to compute, and are listed in lines $D1$ and $D2$ in Table 1. We synthesize these metrics to measure a network's *hardware heterogeneity* using a normalized entropy metric (line $D3$): $\frac{-\sum_{i,j} p_{ij} log_2 p_{ij}}{log_2 N}$, where p_{ij} is the fraction of devices of model i that play role j (e.g., switch, router, firewall, load balancer) in the network, and N is the size of the network. This metric captures the extent to which the same hardware model is used in multiple roles, or multiple models are used in the same role; a value close to 1 indicates significant heterogeneity. We compute a similar firmware heterogeneity metric.

Computing metrics that capture the logical composition and structure of the data and control planes is more intricate as it involves parsing configuration files. To conduct our study, we extended Batfish [11] to parse the configuration languages of various device vendors (e.g., Cisco IoS). Given parsed configurations, we determine the logical composition of the data plane by enumerating the number of logical data plane constructs used (e.g., spanning tree, VLAN, link aggregation), as well as the number of instances of each (e.g., number of VLANs configured); Table 1, line $D4$.

To model control plane structure, we leverage prior work on configuration models [5]. In particular, we extract routing instances from device configurations, where each instance is a collection of routing processes of the same type (e.g., OSPF processes) on different devices that are in the transitive closure of the "adjacent-to" relationship. A network's routing instances collectively implement its control plane. We enumerate the number of such instances, as well as the average size of each instance (Table 1, line $D5$) using the same methodology as Benson et al. [5].

Finally, we enumerate the average number of inter- and intra-device configuration references in a network [5]. These metrics (Table 1, line $D6$) capture the configuration complexity imposed in aggregate by all aspects of a network's design, as well as the impact of specific configuration practices followed by operators.

Operational Practices. We infer operational practices by comparing two successive configuration snapshots from the same device. If at least one stanza differs, we count this as a configuration change.

We compute basic statistics about the configuration changes observed over a certain time window (Table 1, line $O1$). In addition, we study the modality of changes (line $O2$). We infer modality (automated vs. manual) using the login metadata stored with configuration snapshots: we mark a change as *automated* if the login is classified as a special account in the organization's user management system. Otherwise we assume the change was *manual*. This conservative approach will misclassify changes made by scripts executing under a regular user account, thereby under-estimating the extent of automated changes.

To model change type, we leverage the fact that configuration information is arranged as *stanzas*, each containing a set of options and values pertaining to a particular construct—e.g., a specific interface, VLAN, routing instance, or ACL. A stanza is identified by a type (e.g., interface) and a name (e.g., TenGigabit0/1). When part (or all) of a stanza is added, removed, or updated, we say a change of type T occurred, where T is the stanza type. We count the number of changes of each type over a certain time window (Table 1, line $O3$).

There are a few challenges and limitations with this approach. First, type names differ between vendors: e.g., an ACL is defined in Cisco IoS using an `ip access-list` stanza, while a `firewall filter` stanza is used in Juniper JunOS. We address this by manually identifying stanza types on different vendors that serve the same purpose, and we convert these to a vendor-agnostic type identifier. Second, even after generalizing types, a change with the same effect may be typified differently on different vendors: e.g., an interface is assigned to a VLAN in Cisco IoS using the `switchport access vlan` option within an `interface` stanza, while in Juniper JunOS the `interface` option is used within a `vlan` stanza; even though the effect of the change is the same, it will be typified as an interface change on a Cisco device and a VLAN change on a Juniper device. Operators using MPA should be aware of this limitation and interpret prediction results according to the mix of vendors in their networks.

In addition to computing change metrics over changes on individual devices, we compute change metrics over *change events* (Table 1, line $O4$). Change events account for the fact that multiple devices' configurations may need to be changed to realize a desired outcome. For example, establishing a new layer-2 network segment (e.g., a VLAN) requires configuration changes to all devices participating in the segment.

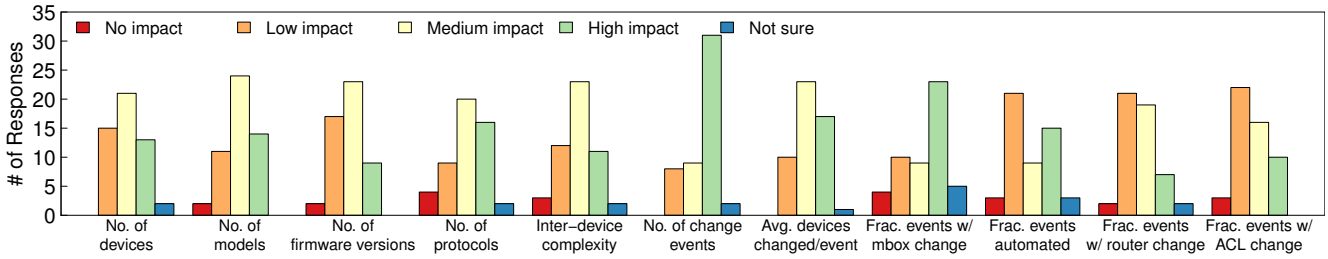


Figure 2: Results of operator survey

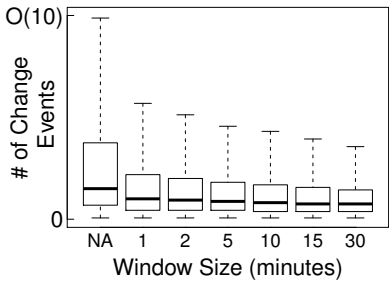


Figure 3: Impact of change grouping threshold (δ) on the number of change events; each box shows the 25th, 50th, and 75th percentile number of change events per-network per-month using various values of δ; whiskers indicate the most extreme datapoints within twice the interquartile range

To identify change events, we group changes using a simple heuristic: if a configuration change on a device occurs within δ time units of a change on another device in the same network, then we assume the changes on both devices are part of the same change event. Figure 3 shows how different values of δ influence the number of change events. The rest of our analysis uses $\delta = 5$ minutes, because operators indicated they complete most related changes within such a time window. In the future, we plan to also consider the change type and affected entities (e.g., VLAN or subnet) to more finely group related changes.

Network Health. The health of a network can be analyzed from many perspectives, including performance (e.g., latency or throughput), quality of experience (e.g., application responsiveness), and failure rate (e.g., packet loss or link/device downtime). Networks are often equipped with monitoring systems that track these metrics and raise alarms when critical thresholds are crossed. In the networks we study, trouble tickets are automatically created when such alarms are raised. Tickets are also created when users report problems or operators conduct planned maintenance. We exclude the latter from our analysis, because maintenance tickets are unlikely to be triggered by performance or availability problems.

Given that ticket logs capture a wide-range of network issues, operators view tickets as a valuable measure of network health. In particular, operators from the OSP indicated that number of tickets is a useful metric. Other metrics computed from network tickets (e.g., number of high impact problems, mean time to resolution, etc.) are less useful because of inconsistencies in ticketing practices: e.g., impact levels are often subjective, and tickets are sometimes not marked as resolved until well after the problem has been fixed. As future work, we plan to explore how to accurately obtain more fine-grained health measures using tools like NetSieve [26].

Property	Value
Months	17, Aug 2013 – Dec 2014
Networks	850+
Services	O(100)
Devices	O(10K)
Config snapshots	O(100K), ≈450GB
Tickets	O(10K), ≈80MB

Table 2: Size of datasets

3. MANAGEMENT PRACTICES TODAY

Today, there is little consensus in the community on the impact of different management practices on networks' health. We conducted two studies that demonstrate this lack of consensus. One study is qualitative in which we surveyed network operators regarding which practices mattered for network health. The other is quantitative in which we systematically characterized management practices in use at a large OSP. The diversity in opinions and actual practices uncovered by these studies motivate the need for MPA. We describe these studies below.

3.1 Operators' Perspectives

Our operator survey covered 51 network operators, whom we recruited through the NANOG mailing list (45 operators), from our campus network (4), and from the large OSP (2). For ten of the practices in Table 1, we asked operators how much they thought each practice would matter to their networks' health. Results of our survey are summarized in Figure 2. We see clear consensus in just one case—number of change events. Otherwise, we note a general diversity of opinion: for several practices—e.g., network size, number of models used, and inter-device configuration complexity—the fractions of operators who said the practice has low vs. high impact are roughly the same. We observed similar diversity even among operators of the OSP and those of the campus network. A handful of operators indicated that they are unsure of the impact of certain management practices.

We also asked operators to "write-in" other practices they believe to have a high impact on network health. The responses included: number of operators, skill levels of operators, documentation and training provided, extent of auditing tools, and extent of pre-change analysis. Unfortunately, these metrics are difficult to quantify, because they relate to the operators (e.g., skill level) rather than the networks. We leave the integration of such metrics into MPA as future work.

3.2 Management Practices in a Large OSP

The diversity in opinions uncovered by the survey could be due to the fact that the operators manage different networks. It is possible that operators know which practices impact their own network, and it is just that this set of practices differs across organizations but is internally consistent. However, our characterization of management practices inside an OSP suggests this is not the case; we find

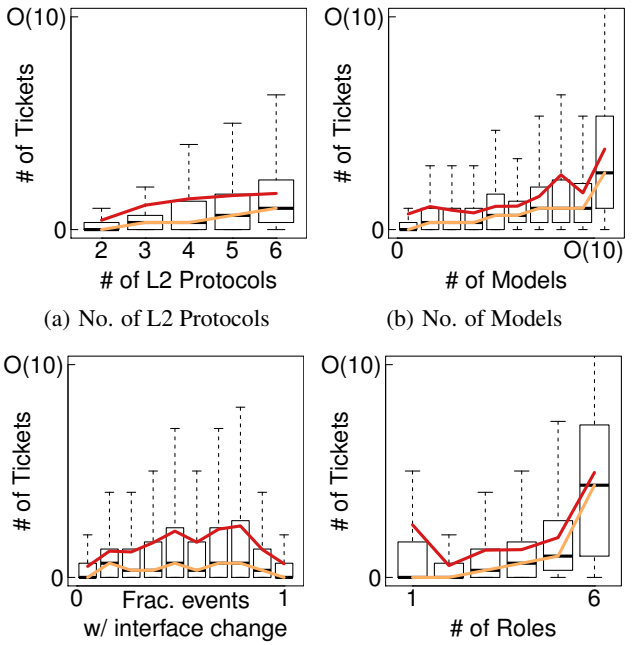
significant heterogeneity in the practices inside the OSP. While detailed characterization is presented in Appendix A, we summarize example findings here.

The OSP owns 850+ networks that are managed based on documented "best practices." Each network hosts one or more Web services or interconnects other networks. Our datasets cover a 17 month period from August 2013 through December 2014. Table 2 shows its key aspects. For confidentiality, we do not list exact numbers for several measures.

Design Practices. We find the control and data planes of networks to be quite heterogeneous in their physical and logical structure. For instance, while the median network's hardware and firmware heterogeneity is low (entropy metric < 0.3), there are several networks ($\approx 10\%$) that are highly heterogeneous on both fronts (metric > 0.67) (Figure 11(a)). Likewise, the number of data and control plane protocols configured across networks is distributed almost uniformly between 1 and 8 (Figure 11(b)). Perhaps most interestingly the overall configuration complexity metrics (intra- and inter-device references) vary by 1-2 orders of magnitude across networks (Figure 11(d)). Per the operator survey (Figure 2), many of these metrics may have a modest impact on health.

Operational Practices. We similarly find significant diversity in what and how networks are changed. For instance, changes to router stanzas are not as common for the median network (where 5% of changes are to router stanzas), but such changes are quite prevalent (>0.5 of all changes) in about 5% of the networks (Figure 12(c)). Likewise, number of change events and changes involving middleboxes—both of which were considered impactful by operators (Figure 2)—show significant diversity: e.g., change event count in the 10th vs 90th percentile network is 3 vs 34 (Figure 12(e)). The modality of changes is also diverse: in 40% of the networks at least half the changes are automated, but in 10% of the networks only 15% of changes are automated (Figure 12(d)).

This level of diversity within the networks of the same organization suggests that operators have little agreement on which practices are good. This lack of agreement is confirmed by our conversation with the operators of the OSP. These conversations also confirm that the operators do not have a way to map adjustments in management practices to shifts in network health. Helping such operators is the goal of MPA.



(a) No. of L2 Protocols (b) No. of Models

(c) Frac. of events with interface change (d) No. of Roles

Figure 4: Tickets based on management practices; boxes show 25th & 75th %tile, while whiskers show 2x the interquartile range; red (dark) line shows the average number of tickets, while orange (light) line shows the median

4. MPA OVERVIEW

Given the diversity and complexity of management practices, we need a systematic framework to understand their impact on network health, and in turn improve management practices. MPA is one such framework.

MPA has two goals. The first is to help operators derive the top k management practices that impact the health (e.g., problem incidence) of their networks. Armed with this information, operators can develop suitable best practices to improve organization-wide design and operational procedures. The main challenges here are that: (i) management practices can differ significantly in the nature of their relationship with network health; and (ii) many practices are often related, impacting both one another and network health. Thus, systematically distilling the "heavy hitters" is not easy. In Section 5, we show how to overcome these challenges by using mutual information to uncover statistical dependencies, and nearest neighbor matching of propensity scores, in the context of quasi-experimental designs (QEDs), to identify causal relationships.

The second goal of MPA is to help operators predict, in an ongoing fashion, what impact the current set of management practices have on the health of *individual* networks. This goes beyond focusing on the top practices; it incorporates the effects of one-off deviations from established procedures, as well as the effects of management practices whose impact on network health manifests only in a narrow set of situations. Armed with such metrics, operators can closely monitor networks that are predicted to have more problems and be better prepared to deal with failures. The main challenge is drawing meaningful conclusions despite limited data for individual networks, especially data for "unhealthy" networks which is often the minority. In Section 6, we show how to overcome this challenge, and build models that accurately predict the health of individual networks, using boosting and oversampling of unhealthy network data.

5. MANAGEMENT PRACTICES THAT IMPACT NETWORK HEALTH

Identifying management practices that impact network health is valuable to operators, yet non-trivial to accomplish. The nature of management practices is such that we face at least two challenges. First, practices may not have a linear, or even monotonic, relationship with network health; this makes it difficult to clearly identify *statistical dependencies*. For example, Figure 4 shows three different management practices—*number of L2 protocols*, *number of models*, and *fraction of events with an interface change*—that have a linear, monotonic, and non-monotonic relationship, respectively, with number of tickets. Second, management practices are often related, such that a change in one practice impacts another practice, as well as network health. For example, Figures 4 and 5 show that *number of models* and *number of roles* are related to network health, and each other. This makes it challenging to identify *causal relationships*.

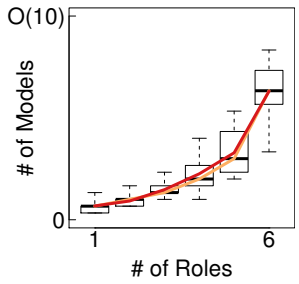


Figure 5: Relationship between *number of models* and *number of roles*

In this section, we present the techniques we use to overcome these challenges. We illustrate how they work using our data from the OSP.

5.1 Dependence Analysis

Common approaches for decomposing the impact of different factors include analysis of variance (ANOVA) [3] and principal/independent component analyses (PCA/ICA) [9]. However, these techniques make key assumptions about underlying dependencies that make them inapplicable to MPA. ANOVA assumes linear relations, which may not always hold, as illustrated earlier in Figure 4. ICA attempts to express the outcome (health metric) as a linear or non-linear combination of independent components; applying PCA first helps identify the components to feed to ICA. The main issue is that the components output by PCA are linear combinations of a subset of management practices. Thus, this approach also makes the implicit assumption that linear combinations of practice metrics can explain network health. Furthermore, the outcome of ICA may be hard to interpret (especially if it relies on a non-linear model).

Instead, we identify statistical dependencies using a more general approach: *mutual information* (MI). When computed between a management practice metric and network health, MI measures how much knowing the practice reduces uncertainty about health. Crucially, MI does not make assumptions about the nature of the relationship.

5.1.1 Mutual Information

The MI between variables X and Y (a management practice and network health) is defined as the difference between the entropy of Y, $H(Y)$, and the conditional entropy of Y given X, $H(Y|X)$. Entropy is defined as $H(Y) = -\sum_i P(y_i) log P(y_i)$, where $P(y_i)$ is the probability that $Y = y_i$. Conditional entropy is defined as $H(Y|X) = \sum_{i,j} P(y_i, x_j) log \frac{P(x_j)}{P(y_i, x_j)}$, where $P(y_i, x_j)$ is the probability that $Y = y_i$ and $X = x_j$. MI is symmetric.

We also examine statistical dependencies between management practices using *conditional mutual information* (CMI). The CMI between a pair of management practices and network health measures the expected value of the practices' MI, given health.[1] The CMI for two variables X_1 and X_2 relative to variable Y is defined as $H(X_1|Y) - H(X_1|X_2, Y)$. Like MI, CMI is also symmetric (with respect to X_1 and X_2).

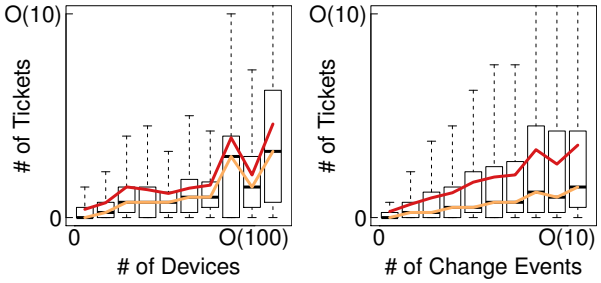
Binning. Prior to computing MI or CMI, we compute the mean value of each management practice and health metric on a monthly basis for each network, giving us ≈11K data points. We bin the data for each metric using 10-equal width bins, with the 5th percentile value as the lower bound for the first bin, and the 95th percentile value as the upper bound for the last bin. Networks whose metric value is below the 5th (above the 95th) percentile are put in the first (last) bin.

Our motivation for this binning strategy is twofold. First, in our characterization of management practices (Appendix A), we observe that many management practices have a long tail: e.g., number of VLANs (Figure 11(c)). Using the 5th and 95th percentile bounds for the first and last bins significantly reduces the range of values covered by each bin, thereby reducing the likelihood that the majority of networks will fall into just one or two bins. Second, minor deviations in a management practice or health metric—e.g., one more device or one more ticket—are unlikely to be significant. Our binning helps reduce noise from such minor variations

[1] In a sense, the pair's joint probability distribution.

Management Practices	Avg. Monthly MI
No. of devices (D)	0.388
No. of change events (O)	0.353
Intra-device complexity (D)	0.329
No. of change types (O)	0.328
No. of VLANs (D)	0.313
No. of models (D)	0.273
No. of roles (D)	0.221
Avg. devices changed per event (O)	0.215
Frac. events w/ interface change (O)	0.201
Frac. events w/ ACL change (O)	0.198

Table 3: Top 10 management practices related to network health according to average monthly MI; parenthetical annotation indicates practice category (D=design, O=operational)



(a) No. of Devices (b) No. of Change Events

Figure 6: Tickets based on management practices

5.1.2 Results for the OSP

We now present statistical dependence results for the OSP. Table 3 shows the 10 management practices that have the strongest statistical dependence with network health. It includes five *design practices* and five *operational practices*, thus highlighting the potential importance of both types of practices to a healthy network.

We visually confirmed the assessment that the practices in Table 3 have a statistical dependence with network health. For example, Figure 6 illustrates the strong statistical dependence with network health for the top two practices. Likewise, Figure 4, which we described earlier, illustrates the relationships for *number of models*, *number of roles*, and *fraction of change events with an interface change* (ranked 6^{th}, 7^{th}, and 9^{th}, respectively, in Table 3).

Interestingly, one of the practices which has high impact according to the survey (Figure 2)—*fraction of events with a middlebox change*—does not appear in the top 10 practices (Table 3); this practice is ranked 23 out of 28. This may be due to the fact that the majority of changes to the OSP's middleboxes are simple adjustments to the server pools configured on load balancers.

Table 4 shows the 10 management practices that have the strongest statistical dependence with *each other*. We observe that six of the top 10 practices related to network health (Table 4) are statistically dependent with other practices; this includes all five of the top

Management Practice Pair		CMI
Frac. events w/ pool change (O)	Frac. events w/ mbox change (O)	1.107
Firmware entropy (D)	Hardware entropy (D)	0.978
No. of OSPF instances (D)	No. of L3 protocols (D)	0.923
No. of models (D)	No. of change types (O)	0.735
No. of BGP instances (D)	Inter-device complexity (D)	0.732
No. of roles (D)	No. of models (D)	0.713
No. of BGP instances (D)	No. of L2 protocols (D)	0.601
Avg. size of an OSPF instance (D)	No. of change types (O)	0.576
Intra-device complexity (D)	Inter-device complexity (D)	0.574
No. of devices (D)	No. of VLANs (D)	0.569

Table 4: Top 10 pairs of statistically dependent management practices according to CMI; highlighted practices are in the top 10 according to MI; parenthetical annotation indicates practice category (D=design, O=operational)

design practices and one operational practice. In general, more design practices (as opposed to operational practices) are statistically dependent with each other. This trend stems from the natural connections between many design decisions: e.g., configuring more BGP instances results in more references between devices and increases inter-device complexity.

We also observe from Table 4 that several practices are dependent with multiple other practices: e.g., *number of models* is dependent with *number of roles* and *number of change types*, and *number of change types* is also dependent with *average size of an OSPF instance*. Thus, evaluating the impact of a management practice on network health requires accounting for many other practices; we next discuss how we achieve this.

5.2 Causal Analysis

Although we can select the k practices with the highest MI as the top k management practices associated with network health, there is no guarantee these practices actually impact health. To establish a *causal relationship* between a management practice and network health, we must eliminate the effects of *confounding factors* (i.e., other practices) that impact this practice and network health [18]. Figures 4 and 5, discussed earlier, illustrate such an effect.

Ideally, we would eliminate confounding factors and establish causality using a *true randomized experiment*. In particular, we would ask operators to employ a specific practice (e.g., decrease the number of device models) in a randomly selected subset of networks; we would then compare the network health (outcome) across the selected (treated) and remaining (untreated) networks. Unfortunately, conducting such experiments takes time (on the order of months), and requires operator compliance to obtain meaningful results. Moreover, true experiments ignore already available historical network data.

To overcome these issues, we use *quasi-experimental design* (QED) [30]. QED uses existing network data to affirm that an independent (or treatment) variable X has a causal impact on a dependent (or outcome) variable Y.

5.2.1 Matched Design

We use a specific type of QED called the *matched design* [33]. The basic idea is to pair cases—each case represents a network in a specific month—that have equal (or similar) values for all confounding variables $Z_1 ... Z_n$, but different values for the treatment variable X. Keeping the confounding variables equal negates the effects of other practices on the outcome (network health), and increases our confidence that any difference in outcomes between the paired cases must be due to the treatment (practice under study).

Using a matched design to identify a causal relationship between a management practice and network health entails four key steps: (1) determine the practice metric values that represent treated and untreated; (2) match pairs of treated and untreated cases based on a set of confounding factors, a distance measure, and a pairing method; (3) verify the quality of the matches to ensure the effect of confounding practices is adequately accounted for; and (4) analyze the statistical significance of differences in outcomes between the treated and untreated cases to determine if there is enough support for a causal relationship.

A key challenge we face in using a matched design is obtaining a sufficient number of quality matches to provide an adequate foundation for comparing the outcomes between treated and untreated cases. As shown in Appendix A, practices tend to vary significantly across networks. Furthermore, many management practices are statistically dependent with network health and each other (Section 5.1). We use nearest neighbor matching based on propensity scores [33] to partially addresses this challenge, but there are also fundamental limitations imposed by the size of our datasets.

We now describe the analysis steps in more detail, using *number of change events* as an example management practice for which we want to establish a causal relationship with network health. At the end of the section, we present results for the 10 management practices that have the highest statistical dependence with network health for the OSP (Table 3).

5.2.2 Determining the Treatment

While most other studies that use QEDs (e.g., those in the medical and social sciences) have a clear definition of what constitutes "treatment," there is no obvious, definitive choice for most management practices. The majority of our management practice metrics have an (unbounded) range of values, with no standard for what constitutes a "normal range": e.g., for the OSP's networks, the average number of change events per month ranges from 0 to hundreds (Figure 12(e)). Hence, we must decide what value(s) constitute *treated* and *untreated*.

One option is to define untreated as the practice metric value that represents the absence of operational actions (e.g., no change events), or the minimum possible number of entities (e.g., one device model or one VLAN). However, we find it is often the case that: (i) several confounding practices will also have the value 0 or 1 (or be undefined) when the treatment practice has the value 0 or 1—e.g., when *number of change events* is 0, *number of change types*, *average devices changed per event*, and *fraction of events with a change of type T* are undefined; and (ii) several confounding practices will be non-zero (or >1) when the treatment practice is non-zero. This observation makes sense, given that our CMI results showed a strong statistical dependence between many management practices (Table 4). Unfortunately, it makes it difficult to find treated cases with similar confounding practices that can be paired with the untreated cases.

Comp. Point	Untreated Cases	Treated Cases	Pairs	Untreated Matched	Abs. Std. Diff. of Means	Ratio of Var.
1:2	8259	1745	1742	1109	0.0000	1.0091
2:3	1745	616	614	431	-0.0002	1.0314
3:4	626	296	295	200	0.0052	1.0744
4:5	296	783	673	174	-0.0002	1.0411

Table 5: Matching based on propensity scores

Given the absence of a "normal range," and the strong statistical dependence between practices, we choose to use multiple definitions of treated and untreated and conduct multiple causal analyses. In particular, we use the same binning strategy discussed in Section 5.1 to divide cases into 5 bins based on the value of the treatment practice. Then we select one bin (b) to represent untreated, and a neighboring bin ($b+1$) to represent treated. This gives us four points of comparison: bin 1 (untreated) vs. bin 2 (treated), 2 vs. 3, 3 vs. 4, and 4 vs. 5; we denote these experimental setups as *1:2*, *2:3*, *3:4*, and *4:5*, respectively. More (or fewer) bins can be used if we have an (in)sufficient number of cases in each bin. In Section 5.2.4, we discuss how to evaluate the quality of matches, which can help determine whether more (fewer) bins can be used.

5.2.3 Matching Pairs of Cases

Matching each treated case with an untreated case is the next step in the causal analysis process. For our causal conclusions to be valid, we must carefully select the confounding factors, distance measure, and pairing method used in the matching process.

During the matching process, it is important to consider all practices (except the treatment practice) that may be related to the treatment or outcome. Excluding a potentially important confounding practice can significantly compromise the validity of the causal conclusion, while including practices that are actually unassociated with the outcome imposes no harm—assuming a sufficiently large sample size and/or a suitable measure of closeness [32]. Therefore, we include all 28 of the practice metrics we infer, minus the treatment practice, as confounding factors.

One caveat of including many confounding practices is that it becomes difficult to obtain many *exact matches*—pairs of cases where both cases have the exact same values for all confounding practices. For example, exact matching produces at most 17 pairs (out of ≈11K cases) when *number of change events* is the treatment practice. The same issue exists when matching based on Mahalanobis distance [29].

We overcome this challenge using *propensity scores*. A propensity score measures the probability of a case receiving treatment (e.g., having a specific *number of models*) given the observed confounding practices (e.g., *number of roles*) for that case [33]. By comparing cases that have the same propensity scores—i.e., an equally likely probability of being treated based on the observed confounding practices—we can be confident that the actual presence or absence of treatment is not determined by the confounding practices. In other words, a treated case and an untreated case with the same propensity score have the same probability of having a given value for a confounding practice (e.g., *number of roles*); thus propensity score matching mimics a randomized experiment.

Given propensity scores for all treated and untreated cases, we use the most common, and simplest, pairing method: *k=1 nearest neighbor* [32]. Each treated case is paired with an untreated case that results in the smallest absolute difference in their propensity scores. To obtain the best possible pairings, we match *with replacement*—i.e., allow multiple treated cases to be paired with the same untreated case. We also follow the common practice of discarding treated (untreated) cases whose propensity score falls outside the range of propensity scores for untreated (treated) cases.

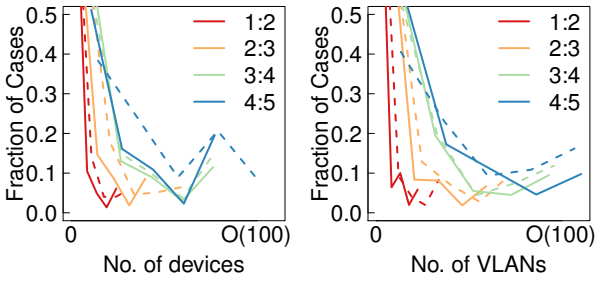


Figure 7: Visual equivalence of confounding practice distributions; lines of the same color are for the same comparison point; solid lines are for matched untreated cases and dashed lines are for matched untreated cases

Table 5 shows the matching results for each of the four comparison points for *number of change events*. There are significantly more matched pairs using propensity scores: up to 99.8% of treated cases are matched, versus <1% with exact matching. Furthermore, the number of untreated cases that are matched with treated cases is less than the number of matched pairs, implying that matching with replacement is beneficial.

5.2.4 Verifying the Quality of Matches

When matching based on propensity scores, rather than the raw values of confounding practices, it is important to verify that the distribution of values for each confounding practice is similar for both the matched treated cases and the matched untreated cases. Otherwise, the effects of confounding practices have not been successfully mitigated, and any causal conclusions drawn from the matched pairs may not be valid.

Figure 7 visually confirms the distribution equivalence for two of the confounding practices. However, to facilitate bulk comparison, we use two common numeric measures of balance: standardized difference of means and ratio of variances [32]. The former is computed as $\frac{\bar{Z}_T - \bar{Z}_U}{\sigma_T}$, where $\bar{Z}_T$ and $\bar{Z}_U$ are the means of a confounding practice Z for the matched treated and matched untreated cases, respectively, and σ_T and σ_U are the standard deviations. The ratio of variances is computed as σ_T^2/σ_U^2. For each confounding practice, the absolute standardized difference of means should be less than 0.25 and the variance ratio should be between 0.5 and 2 [32]. These equations and thresholds also apply to the propensity scores for the matched cases.

As shown in Table 5, the absolute standard difference of means and the ratio of variances of the propensity scores satisfy the quality thresholds for all comparison points. The same also holds for all confounding factors (not shown).

Although we consider a large set of management practices in our causal analysis, it is possible that other practices or factors also contribute to the observed outcomes. We can easily incorporate new practices into our propensity scores as we learn about them. Additionally, our matching based on propensity scores introduces some randomness that can help mitigate the effects of any unaccounted for factors. However, we can never definitely prove causality with QEDs [21]; any causal relationships identified by MPA should thus be viewed as "highly-likely" rather than "guaranteed".

5.2.5 Analyze the Statistical Significance

The final step is to analyze the statistical significance of the difference in outcomes between the matched treated and untreated cases. For each matched pair, we compute the difference in outcome (*number of tickets*) between the treated and untreated case: $y_t - y_u$. If the result is positive (negative), then the treatment prac-

Comparison Point	Fewer Tickets	No Effect	More Tickets	p-value
1:2	562	350	830	6.80×10^{-13}
2:3	251	61	302	3.34×10^{-2}
3:4	110	25	160	2.80×10^{-3}
4:5	282	38	343	1.63×10^{-2}

Table 6: Statistical significance of outcomes; causality is deemed to exist for higlighted comparison points

tice has led to worse (better) network health; if the result is zero, then the practice has not impacted health. We use the outcome calculations from all pairs to produce a binomial distribution of outcomes: more tickets (+1) or fewer tickets (-1). Table 6 shows the distribution for the matched pairs at each comparison point.

If the treatment practice impacts network health, we expect the median of the distribution to be non-zero. Thus, to establish a causal relationship, we must reject the null hypothesis H_0 that the median outcome is zero. We use the *sign test* to compute a *p-value*—the probability that H_0 is consistent with the observed results. Crucially, the sign test makes few assumptions about the nature of the distribution, and it has been shown to be well-suited for evaluating matched design experiments [15]. We choose a moderately conservative threshold of 0.001 for rejecting H_0.

Table 6 shows the p-value produced by the sign test for each of the comparison points for *number of change events*. We observe that the p-value is less than our threshold for the *1:2* comparison point. Hence, the difference in the *number of change events* between bins 1 and 2 is statistically significant, and a causal impact on network health exists at these values. In contrast, the results for the other comparison points (*2:3*, *3:4*, and *4:5*) are not statistically significant. This is due to either the absence of a causal relationship—i.e., increasing the *number of change events* beyond a certain level does not cause an increase in the *number of tickets*—or an insufficient number of samples. We believe the latter applies for our data, because there is at least some evidence of a non-zero median: the number of cases with more tickets is at least 20% higher than the number of cases with fewer tickets for the *2:3*, *3:4*, and *4:5* comparison points.

5.2.6 Results for the OSP

We now conduct a causal analysis for the 10 management practices with the highest statistical dependence with network health (Table 3). Due to skew in our data, we can only draw meaningful conclusions for low values of our practice metrics (bins 1 and 2).

Table 7 shows the p-value for the comparison between the first and second bin for each practice. We observe that 8 of the 10 practices have a causal relationship according to our p-value threshold. In fact, the p-values for these practices are well below our chosen threshold (0.001). Furthermore, several of the practices with a causal relationship, including *number of devices* and *average devices changed per event*, are practices for which operators had mixed opinions regarding their impact (Figure 2). Our analysis also matches the prevailing opinion that *number of change events* has a high impact on health, and, to some extent, discredits the belief that the *fraction of events with ACL changes* has low impact.

For the remaining two metrics, *intra-device complexity* and *fraction of events with an interface change*, there is not enough evidence to support a causal relationship. The high statistical dependence but lack of a causal relationship is likely due to these practices being affected by other practices which do have a causal relationship with network health. For example, *number of VLANs* has a causal relationship with network health and may influence *intra-device complexity*. Hence, a change in *number of VLANs* may change both network health and *intra-device complexity* in a way that makes *intra-device complexity* statistically similar to health.

Treatment Practice	p-value for 1:2
No. of devices	1.92×10^{-8}
No. of change events	1.05×10^{-12}
Intra-device complexity	1.53×10^{-2}
No. of change types	5.75×10^{-12}
No. of VLANs	6.46×10^{-6}
No. of models	1.31×10^{-7}
No. of roles	2.99×10^{-10}
Avg. devices changed per event	3.56×10^{-8}
Frac. events w/ interface change	5.27×10^{-3}
Frac. events w/ ACL change	9.10×10^{-9}

Table 7: Causal analysis results for the first and second bin for the top 10 statistically dependent management practices; highlighted p-values satisfy our significance threshold

Treatment Practice	Comparison Point 2:3	3:4	4:5
No. of devices	Imbal.	Imbal.	Imbal.
No. of change events	3.34×10^{-2}	2.80×10^{-3}	1.63×10^{-2}
Intra-device complexity	Imbal.	1.71×10^{-1}	1.47×10^{-1}
No. of change types	9.02×10^{-1}	1.42×10^{-5}	Imbal.
No. of VLANs	Imbal.	1.94×10^{-3}	Imbal.
No. of models	Imbal.	Imbal.	Imbal.
No. of roles	Imbal.	6.63×10^{-1}	Imbal.
Avg. devices changed per event	4.53×10^{-3}	2.25×10^{-1}	Imbal.
Frac. events w/ interface change	4.51×10^{-2}	4.58×10^{-1}	2.89×10^{-12}
Frac. events w/ ACL change	4.88×10^{-2}	2.78×10^{-1}	6.48×10^{-2}

Table 8: Causal analysis results for upper bins for the top 10 statistically dependent management practices; highlighted p-values satisfy our significance threshold

Table 8 shows the p-value for the comparison between the upper bins for the same 10 practices. We observe that over one-third of the matchings have poor quality (i.e., strong imbalance), and most of the others have large p-values. This primarily stems from management practice metrics following a heavy-tailed distribution. For example, when the treatment practice is *number of devices*, 81% of cases fall into the first bin and 8% fall into the second bin; this means there are few cases from which to select matched pairs for the *2:3*, *3:4*, and *4:5* comparison points. The only way to address this issue is to obtain (more diverse) data from more networks.

Acting on the Results. The ability to change practices that cause poor network health varies based on the class of practice and the needs of the organization. Changing design practices (e.g., number of models or roles) requires deploying new networks, or significantly overhauling existing networks. In contrast, operational practices can be adjusted more easily: e.g., changes can be aggregated or reduced by more carefully planning network reconfigurations. Some practices may be difficult to change due to workload demands (e.g., number of devices), but operators can still benefit from understanding these relationships to aid in setting service level objectives (SLOs) or making staffing decisions.

6. PREDICTING NETWORK HEALTH

We now move on to the second goal of MPA: building models that take a network's current management practices as input and predict the network's health. Such models are useful for network operators to explore how adjustments in management practices will likely impact network health: e.g., will combining configuration changes into fewer, larger changes improve network health?

We find that basic learning algorithms (e.g., C4.5 [27]) produce mediocre models because of the skewed nature of management practices and health outcomes. In particular, they over-fit for the

majority healthy network case. Thus, we develop schemes to learn more robust models despite this limitation. We show that we can predict network health at coarse granularity (i.e., healthy vs. unhealthy) with 91% accuracy; finer-grained predictions (i.e., a scale of 1 to 5) are less accurate due to a lack of sufficient samples.

6.1 Building an Organization's Model

We start with the following question: given all data from an organization, what is the best model we can construct?

An intuitive place to start is support vector machines (SVMs). SVMs construct a set of hyperplanes in high-dimensional space, similar to using logistic regression to construct propensity score formulas during causal analysis. However, we found the SVMs performed worse than a simple majority classifier. This is due to unhealthy cases being concentrated in a small part of the management practice space.

To better learn these unhealthy cases, we turn to decision tree classifiers (the C4.5 algorithm [27]). Decision trees are better equipped to capture the limited set of unhealthy cases, because they can model arbitrary boundaries between cases. Furthermore, they are intuitive for operators to understand.

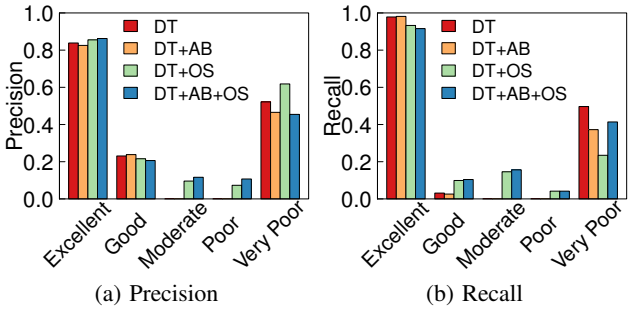
Methodology. Prior to learning, we bin data as described in Section 5.1.1. However, we use only 5 bins for each management practice (instead of 10), because the amount of data we have is insufficient to accurately learn fine-grained models. For network health, we use either 2 bins or 5 bins; two bins (classes) enables us to differentiate coarsely between *healthy* ($\leq$1 tickets) and *unhealthy* networks, while five bins captures more fine-grained classes of health—excellent, good, moderate, poor, and very poor ($\leq$2, 3–5, 6–8, 9–11, and $\geq$12 tickets, respectively). As is standard practice, we prune a decision tree to avoid over-fitting: each branch where the number of data points reaching this branch is below a threshold α is replaced with a leaf whose label is the majority class among the data points reaching that leaf. We set α=1% of all data.

Model Validation. We measure the accuracy, precision, and recall of the decision trees using 5-fold cross validation. *Accuracy* is the mean fraction of test examples whose class is predicted correctly. For a given class C, *precision* measures what fraction of the data points that were predicted as class C actually belong to class C, while *recall* measures what fraction of the data points that belong to class C are correctly predicted as class C.

We find that a 2-class model performs very well. Accuracy of the pruned decision tree—i.e., the mean fraction of test examples whose class is predicted correctly—is 91.6%. In comparison, a majority class predictor has a significantly worse accuracy: 64.8%. Furthermore, the decision tree has very high precision and recall for the healthy class (0.92 and 0.98, respectively), and moderate precision and recall for the unhealthy class (0.62 and 0.31, respectively). A majority class predictor has only moderate precision (0.64) for the healthy class and no precision or recall for the unhealthy class.

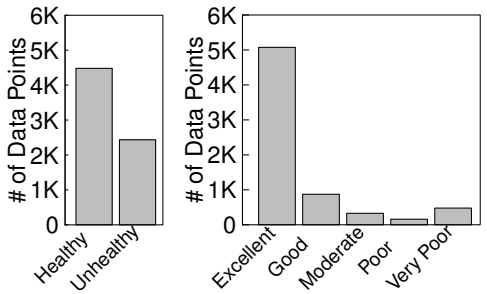
The accuracy for a 5-class model is 81.1%, but the precision and recall for the intermediate classes (good, moderate, and poor) are very low (DT bars in Figure 8). The root cause here is skew in the data: as shown in Figure 9(b), a majority of the samples represent the "excellent health" case (73%), with far fewer samples in other health classes (e.g., the poor class has just 2.3% of the samples). Our 5-class decision tree ends up overfitting for the majority class.

Addressing Skew. Because networks are generally healthy, such skew in data is a fundamental challenge that predictive models in MPA need to address, especially when attempting to predict fine-grained health classes. To address skew and improve the accuracy of our models for minority classes, we borrow two techniques



(a) Precision (b) Recall

Figure 8: Accuracy of 5 class models (DT=standard decision tree learning algorithm, AB=AdaBoost, OS=oversampling)



(a) 2 classes (b) 5 classes

Figure 9: Health class distribution

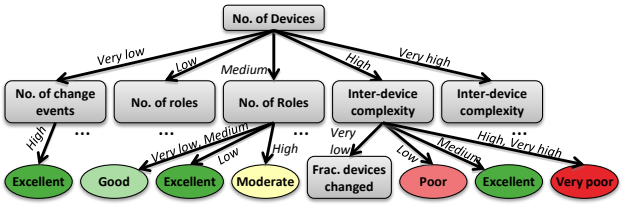
from the machine learning community: boosting (specifically, AdaBoost [12]) and oversampling.[2]

AdaBoost helps improve the accuracy of "weak" learners. Over many iterations (we use 15) AdaBoost increases (decreases) the weight of examples that were classified incorrectly (correctly) by the learner; the final learner (i.e., decision tree) is built from the last iteration's weighted examples. Oversampling directly addresses skew as it repeats the minority class examples during training. When building a 2-class model we replicate samples from the unhealthy class twice, and when building a 5-class model we replicate samples from the poor class twice and the moderate and good classes thrice.

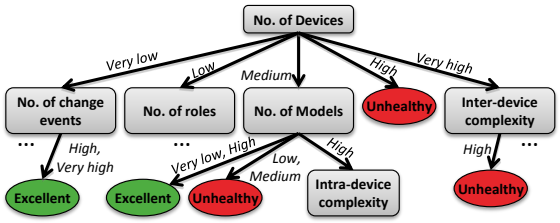
The results from applying these enhancements are shown in Figure 8. We observe that AdaBoost results in minor improvement for all classes. In contrast, using oversampling significantly improves the precision and recall for the three intermediate health classes, and causes a slight drop in the recall for the two extreme classes (excellent and very poor). Using oversampling and AdaBoost in combination offers the best overall performance across all classes.

The final 5-class model is substantially better than using a traditional decision tree. However, it is still sub-optimal due to the significant skew in the underlying dataset. Separating apart a pair of nearby classes whose class boundaries are very close—e.g., excellent and good—requires many more *real* data points from either class; oversampling can only help so much. Thus, lack of data may pose a key barrier to MPA's ability to model network health at *fine granularity*. Nonetheless, we have shown that *good models can be constructed for coarse grained prediction*.

[2]We also experimented with random forests [8, 19]; neither balanced [8] nor weighted random forests [19] improve the accuracy for the minority classes beyond the improvements we are already able to achieve with boosting and oversampling.



(a) 5-class



(b) 2-class

Figure 10: Decision trees (only a portion is shown)

6.2 Using an Organization's Model

Operators can use an organization's model to determine which combinations of management practices lead to an (un)healthy network, and to evaluate how healthy a network will be in the future when a specific set of management practices are applied.

Tree Structure. Figure 10(a) shows a portion of the best 5-class tree. Since decision trees are built by recursively selecting the node with the highest mutual information, the management practice with the strongest statistical dependence (identified in Section 5.1)—*number of devices*—is the root of the tree. In the second level, however, two of the three practices are not present in our list of the 10 most statistically dependent practices (Table 3). This shows that the importance of some management practices depends on the nature of other practices: e.g., when the *number of devices* is medium or low, the *number of roles* is a stronger determinant of health than the *number of change events*. Thus, examining the paths from the decision tree's root to its leaves provides valuable insights into which combinations of management practices lead to an (un)healthy network.

The same observations apply to the 2-class tree (Figure 10(b)).

Predicting Future Health. We now show that an organization's model can accurately predict the future health of an organization's networks. In particular, we build decision trees using data points from M months ($t - M$ to $t - 1$). Then, we use management practice metrics from month t to predict the health of each network in month t. The accuracy for month t is the fraction of networks whose health was correctly predicted.

Table 9 shows the average accuracy for M=1, 3, 6, and 9 for values of t between February and October 2014. We observe that *a 2-class model has consistently high prediction accuracy of 89%* irrespective of the amount of prior data used to train the model. This trend primarily stems from having less severe skew between the majority and minority classes when using two classes (Figure 9(a)).

The prediction accuracy of a 5-class model reaches 78% for $M = 9$. Also, accuracy improves with a longer history of training data: using 9 months, rather than 1 month, of training data results in a 5% increase in accuracy. However, as the amount of training data increases (i.e., increasing M) the relative improvement in accuracy diminishes. Thus, a reasonably accurate prediction of network health can be made with less than a year's worth of data.

M (months)	5 classes	2 classes
1	0.734	0.881
3	0.756	0.893
6	0.779	0.901
9	0.779	0.903

Table 9: Accuracy of future health predictions

7. DISCUSSION

Generality. While the observations we make for the OSP's networks provide a valuable perspective on the relationship between management practices and network health, the statistical dependencies and causal relationships we uncover may not apply to all organizations. Differences in network types (e.g., data center vs. wide area networks), workloads, and other organization-specific factors may affect these relationships. Nonetheless, our techniques are likely generally applicable, and any organization can run our tool [2] to discover these relationships for its networks.

Intent of Management Practices. The metrics we infer (Section 2.2) quantify management practices in terms of their direct influence on the data and control planes: e.g., how heterogeneous is network hardware, and which configuration stanzas are changed. However, we could also quantify management actions in terms of their *intent*, or the goal an operator is trying to achieve: e.g., an operator may want to reduce firmware licensing costs, so they design a network to use RIP rather than OSPF [5]. By analyzing the relationships between intent and network health, we can gain a richer understanding of what practices are the most problematic. Unfortunately, intent is much more difficult to infer from network data sources (Section 2.1), and doing so is part of our ongoing work.

8. RELATED WORK

An earlier version of this work [4] introduced the idea of management plane analytics and provided visual evidence of correlations between a few management practices and network health. This paper considers many more practices, conducts causal analyses, and shows how to build accurate predictive models.

Establishing, following, and refining management practices is an important part of information technology (IT) service management. ITIL [1] provides guidance on: service design, which focuses on health-related concerns such as availability and service levels; service transition, which focuses on change, configuration, and deployment management; and continual service improvement. Some of the general metrics used in ITIL to asses the health of an IT organization (e.g., *number of changes*) are also used in MPA, but MPA also considers may networking-specific metrics. This makes MPA a valuable tool for continual service improvement. The major steps in MPA—defining metrics, characterizing practices, and uncovering relationships between practices and health—are similar to the steps employed in security management [17], but MPA's analyses are focused more on causality and relationship modeling.

A few prior studies have examined network management practices. Kim et al. study several design and operational issues in two campus network: e.g., how network-wide configuration size grows over time, what causes this growth, how configurations of different device types (e.g., router, firewall, etc.) change and why, and the qualitative differences among the campuses in these aspects [20]. Others have looked at more narrow aspects of design and operations: e.g., Benson et al. examine configuration complexity in 7 enterprise networks [5] and study design and change patterns of various network-based services of a large ISP [6]; Garimella et al. and Krothapalli et al. study VLAN usage in a single campus network [13, 23]. In contrast to these prior works, we examine a much more comprehensive set of design and operational practices. Also,

by virtue of studying many networks of a large OSP, we are able to provide a unique view into the *variation* in management practices. Finally, none of the prior studies tie their the observations to health.

Prior work has also examined network health in great detail. For example, Turner et al. use device logs, network probes and incident reports to understand causes and frequency of failures and their impact [37]. Failure in data center networks, and of middleboxes in data centers have been studied by Gill et al. [14] and Navendu et al. [25], respectively. Turner et al. examine how to combine various data sources to obtain a better view of failures and their root causes [38, 39]. However, these studies don't link network issues back to design and operational practices. That said, some of the data sources and techniques considered in these approaches could be valuable to deriving better network health metrics that could then improve MPA.

QEDs have been widely studied. Stuart [32] provides a comprehensive survey of the various techniques employed in matched design experiments. Our use of QEDs is inspired by recent network measurement studies focused on video streaming quality [21] and video ad placement [22]. While these works use exact matching in their QEDs, we use nearest neighbor matching of propensity scores, because exact matching cannot accommodate the large number of confounding factors in the management plane (Section 5.2).

MPA is inspired by how research into software engineering practices, also called "empirical software engineering," has helped improve the quality of software and reduced the number of bugs [7]. We expect similar positive impact from MPA.

9. CONCLUSION

We presented a *management plane analytics* framework for analyzing and improving network management practices. We showed that a systematic analysis of the management plane is: (i) *necessary*—given the diversity in prevalent management practices and in opinions among operators regarding what matters versus not; and (ii) *feasible*—by analyzing data from many networks using carefully selected techniques. We found that the nature of network management data necessitates the use of propensity scores to reduce data dimensionality and facilitate matched design quasi-experiments that identify causal relationships between practices and network health. Additionally, we showed that oversampling and boosting are necessary for building good predictive models in the face of heavily skewed data. Our application of MPA to networks of a large OSP revealed intriguing insights: e.g., the fraction of changes where an ACL is modified has a moderately high impact on network health despite a majority opinion that the impact is low, and the fraction of change events affecting a middlebox has low impact on health despite the belief that the impact is high.

However, we have just scratched the surface of such analysis. There are many issues left open by our work, including: studying other health metrics using MPA, determining how to extend MPA to apply across organizations, and developing tools for inferring management practices from outside a network (akin to probing tools such as trace-route) as opposed to analyzing internally-collected data. We believe this is a rich avenue for future research.

10. ACKNOWLEDGMENTS

We thank the operators of the online service provider for their suggestions and feedback, as well as the operators who took the time to answer our survey. We also thank our shepherd Anja Feldmann and the anonymous reviewers for their insightful comments. This work is supported by the Wisconsin Institute on Software-defined Datacenters of Madison and National Science Foundation grants CNS-1302041, CNS-1330308, and CNS-1345249. Aaron Gember-Jacobson is supported by an IBM PhD Fellowship.

11. REFERENCES

[1] ITIL – IT service management. `http://www.axelos.com/best-practice-solutions/itil`.

[2] Management plane analytics tool. `http://cs.wisc.edu/~agember/go/mpa`.

[3] NIST/SEMATECH e-handbook of statistical methods. `http://itl.nist.gov/div898/handbook`.

[4] A. Akella and R. Mahajan. A call to arms for management plane analytics. In *IMC*, 2014.

[5] T. Benson, A. Akella, and D. Maltz. Unraveling complexity in network management. In *NSDI*, 2009.

[6] T. Benson, A. Akella, and A. Shaikh. Demystifying configuration challenges and trade-offs in network-based ISP services. In *SIGCOMM*, 2011.

[7] C. Bird, B. Murphy, N. Nagappan, and T. Zimmermann. Empirical software engineering at Microsoft Research. In *Computer Supported Cooperative Work (CSCW)*, 2011.

[8] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[9] P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, Apr. 1994.

[10] A. B. Downey. Using pathchar to estimate internet link characteristics. In *SIGCOMM*, 1999.

[11] A. Fogel, S. Fung, L. Pedrosa, M. Walraed-Sullivan, R. Govindan, R. Mahajan, and T. Millstein. A general approach to network configuration analysis. In *NSDI*, 2015.

[12] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *J. Computer and System Sciences*, 55(1):119–139, Aug. 1997.

[13] P. Garimella, Y. Sung, N. Zhang, and S. Rao. Characterizing VLAN usage in an Operational Network. In *SIGCOMM Workshop on Internet Network Management*, pages 305–306, 2007.

[14] P. Gill, N. Jain, and N. Nagappan. Understanding network failures in data centers: Measurement, analysis, and implications. In *SIGCOMM*, 2011.

[15] M. Hollander and D. Wolfe. *Nonparametric statistical methods*. Wiley, 1973.

[16] M. Jain and C. Dovrolis. End-to-end available bandwidth: Measurement methodology, dynamics, and relation with TCP throughput. In *SIGCOMM*, 2002.

[17] A. Jaquith. *Security Metrics: Replacing Fear, Uncertainty, and Doubt*. Addison-Wesley, 2007.

[18] D. D. Jensen, A. S. Fast, B. J. Taylor, and M. E. Maier. Automatic identification of quasi-experimental designs for discovering causal knowledge. In *KDD*, 2008.

[19] T. M. Khoshgoftaar, M. Golawala, and J. Van Hulse. An empirical study of learning from imbalanced data using random forest. In *International Conference on Tools with Artificial Intelligence (ICTAI)*, 2007.

[20] H. Kim, T. Benson, A. Akella, and N. Feamster. The evolution of network configuration: A tale of two campuses. In *IMC*, 2011.

[21] S. S. Krishnan and R. K. Sitaraman. Video stream quality impacts viewer behavior: Inferring causality using quasi-experimental designs. In *IMC*, 2012.

[22] S. S. Krishnan and R. K. Sitaraman. Understanding the effectiveness of video ads: A measurement study. In *IMC*, 2013.

[23] S. D. Krothapalli, X. Sun, Y.-W. E. Sung, S. A. Yeo, and S. G. Rao. A toolkit for automating and visualizing VLAN configuration. In *SafeConfig*, 2009.

[24] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. User-level Internet path diagnosis. In *SOSP*, 2003.

[25] R. Potharaju and N. Jain. Demystifying the dark side of the middle: A field study of middlebox failures in datacenters. In *IMC*, 2013.

[26] R. Potharaju, N. Jain, and C. Nita-Rotaru. Juggling the jigsaw: Towards automated problem inference from network trouble tickets. In *NSDI*, 2013.

[27] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[28] Really Awesome New Cisco ConfIg Differ (RANCID). http://shrubbery.net/rancid.
[29] D. B. Rubin. Using multivariate matched sampling and regression adjustment to control bias in observational studies. *Journal of the American Statistical Association*, 74:318–328, 1979.
[30] W. Shadish, T. Cook, , and D. Campbell. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Houghton Mifflin, 2002.
[31] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson. Measuring ISP topologies with Rocketfuel. *IEEE/ACM Transactions on Networking (ToN)*, 2004.
[32] E. A. Stuart. Matching methods for causal inference: A review and a look forward. *Statistical Science*, 25, 2010.
[33] E. A. Stuart and D. B. Rubin. Best practices in quasi-experimental designs: Matching methods for causal inference. In *Best Practices in Quantitative Methods*, pages 155–176. Sage, 2008.
[34] Y. Sung, S. Rao, S. Sen, and S. Leggett. Extracting network-wide correlated changes from longitudinal configuration data. In *PAM*, 2009.
[35] Traceroute.org. http://www.traceroute.org.
[36] HP OpenView TrueControl Software. http://support.openview.hp.com.
[37] D. Turner, K. Levchenko, J. C. Mogul, S. Savage, and A. C. Snoeren. On failure in managed enterprise networks. Technical Report HPL-2012-101, HP.
[38] D. Turner, K. Levchenko, S. Savage, and A. C. Snoeren. A comparison of syslog and is-is for network failure analysis. In *IMC*, 2013.
[39] D. Turner, K. Levchenko, A. C. Snoeren, and S. Savage. California fault lines: Understanding the causes and impact of network failures. In *SIGCOMM*, 2010.

APPENDIX

A. CHARACTERIZATION OF MANAGEMENT PRACTICES

We provide a detailed characterization of the management practices used at a large online service provider (OSP). This offers a unique and rich view into the practices used in a modern, professionally-managed infrastructure. We are not claiming that this view is representative. For brevity, we quantify a subset of the practice metrics in Table 1. We find significant diversity in the design and operational practices employed across the OSP's networks.

A.1 Design Practices

We start by examining the OSP's networks in terms of their network composition, structure, and purpose.

The majority (81%) of networks host *only one workload*—networks are quite homogeneous in this respect. A handful of networks do not host any workloads; they only connect networks to each other or the external world.

The networks contain a mix of device roles, including routers, switches, firewalls, application delivery controllers (ADCs)[3], and load balancers. Most networks (86%) have devices in multiple roles—although no single device has more than one role—and 71% of networks contain at least one middlebox (firewall, ADC, or load balancer). We also find that over 81% of networks contain devices from more than one vendor, with a maximum of 6, and over 96% of networks contain more than one device model, with a maximum of 25. Thus, some networks must use more than one device model for the same role. Indeed, a closer look at the hardware entropy of the networks (solid line in Figure 11(a)) shows that only 4% of networks have just one model and one role; *the remaining (96% of) networks have varying degrees of heterogeneity*, up to a maximum entropy metric value of 0.82. The extent of firmware heterogeneity is similar (dashed line in Figure 11(a)).

Next, we look at the logical composition and structure of the data and control planes. As shown in Figure 11(b), all networks use at least two layer-2 protocols (VLAN, spanning tree, link aggregation, unidirectional link detection (UDLD), DHCP relay, etc.), and 89% of networks use at least one routing protocol (BGP and/or OSPF). Furthermore, 10% of networks use 8 different protocols. Overall *there is significant diversity in the combination of protocols used*.

We find the same diversity in the number of instances of each protocol. Less than 5 VLANs are configured in 5% of networks, but over 100 VLANs are configured in 9% of networks (Figure 11(c)). Similarly, 86% of networks use BGP for layer-3 routing, with just one BGP instance in 39% of networks and more than 20 instances in 8% of networks (Figure 11(e)). In contrast, only 31% of networks use OSPF for layer-3 routing, with just one or two OSPF instances used in these networks.

Finally, to characterize configuration complexity, Figure 11(d) shows a CDF of intra- and inter-device referential complexity. We find that some networks' configuration is extremely complex (based on Benson et al.'s metrics [5]): in 20% of networks, the mean intra- and inter-device reference counts are higher than 100. However, it is worth noting that: (i) *the range in complexity is rather large*, and (ii) most networks have significantly lower configuration complexity metrics than the worst 10%.

A.2 Operational Practices

We now characterize the frequency, type, and modality of configuration changes, as well as those of change events.

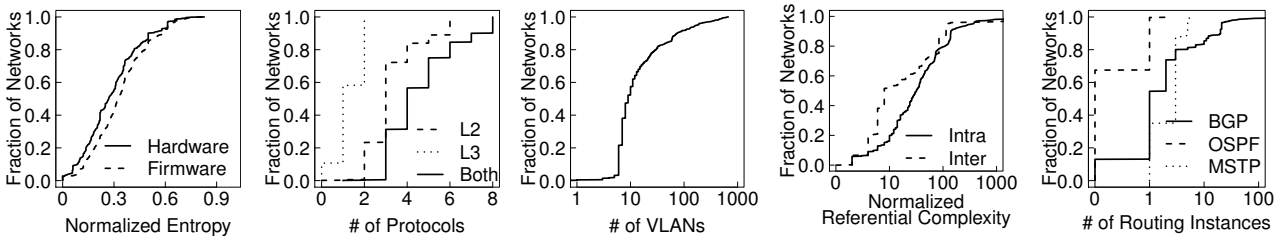
In general, the average number of configuration changes per month is correlated with network size (Figure 12(a); Pearson correlation coefficient of 0.64). However, *several large networks have relatively fewer changes per month*: e.g., one network has over 300 devices but fewer than 150 changes per month. Likewise, *there are several small networks with a disproportionately high change rate.* Furthermore, not every device is changed every month—in 77% of networks less than half of a network's devices are changed in a given month—but most devices are changed at least once per year—in 80% of networks more than three-quarters of the devices are changed in a year (Figure 12(b)). Thus, *changes occur frequently, and to different sets of devices in different months.*

We now analyze different types of changes. Across our entire dataset there are ≈480 different types of changes. Figure 12(c) shows CDFs of the fraction of changes in which at least one stanza of a given type is changed. On a per-network basis, interface changes are the most common, followed by pool (used on load balancers), ACL, user, and router. [4]

Among the above most-frequently changed types, pool changes are also the most frequently automated—more than half of all pool changes are automated in 77% of networks—followed by ACL and interface changes. We also look at the extent of automation over *all* types of changes. As shown in Figure 12(d), more than half (quarter) of the changes each month are automated in 41% (81%) of networks. In general, *we note a significant diversity in the extent of automation*: it ranges between 10% and 70%. Equally interestingly, the fraction of automated changes is not strongly correlated with the number changes (Pearson correlation coefficient is 0.23). Furthermore, the types of changes that are automated most frequently—sflow and QoS—are not the most frequent types of changes.

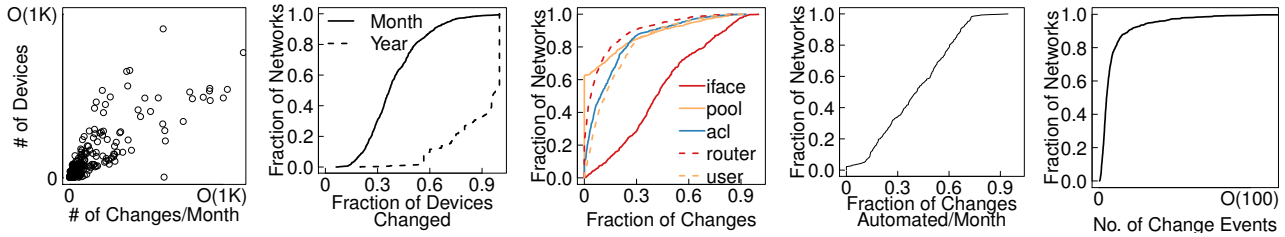
[3]ADCs perform TCP and SSL offload, HTTP compression and caching, content-aware load balancing, etc.

[4]There are no pool changes in 63% of networks because these networks do not contain load balancers.



(a) Device heterogeneity (b) Protocol usage (c) No. of VLANs (d) Referential complexity (e) No. of routing Instances

Figure 11: Characterization of design practices

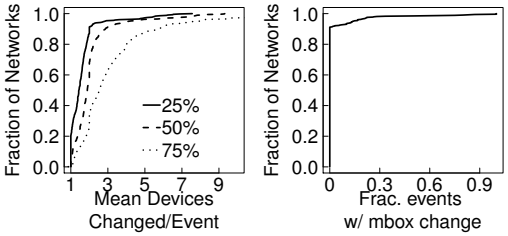


(a) Avg. number of changes per month (b) Avg. fraction of devices changed (c) Most frequent config changes (d) Frac. of changes automated (e) Avg. number of change events per month

Figure 12: Characterization of configuration changes

Lastly, we look at change events, both in terms of how many there are in a network as well as the composition of an event (in terms of number of devices changed). Figure 12(e) shows a distribution of the number of change events. They are few in number ($\mathcal{O}(10)$) in most networks (80%); however about 5% of the networks experience tens if not hundreds of change events in a month. We see a similar *diversity in the number of change events* involving middleboxes (Figure 13(b)). Both prevalence of change events, as well as events involving middleboxes, were flagged by the operators we surveyed as being impactful (Figure 2).

Figure 13(a) shows a CDF of the average number of devices changed per change event. Most change events we see across networks are small: in about half of the networks, a change event affects only one or two devices (on average). Further, in almost all networks, the average change event affects only one device role and one device model. Limiting changes to just a few, similar devices is intuitively a good practice to simplify debugging and rollback.



(a) Avg. devices changed per event (b) Frac. of events that change a middlebox

Figure 13: Characterization of configuration change events