Extracting Network-Wide Correlated Changes from Longitudinal Configuration Data

Yu-Wei Eric Sung[1], Sanjay Rao[1], Subhabrata Sen[2], and Stephen Leggett[3]

[1] Purdue University
[2] AT&T Labs Research
[3] AT&T Inc.

Abstract. IP network operators face the challenge of making and managing router configuration changes to serve rapidly evolving user and organizational needs. Changes are expressed in low-level languages, and often impact multiple parts of a configuration file and multiple routers. These dependencies make configuration changes difficult for operators to reason about, detect problems in, and troubleshoot. In this paper, we present a methodology to extract network-wide correlations of changes. From longitudinal snapshots of low-level router configuration data, our methodology identifies syntactic configuration blocks that changed, applies data mining techniques to extract correlated changes, and highlights changes of interest via operator feedback. Employing our methodology, we analyze an 11-month archive of router configuration data from 5 different large-scale enterprise Virtual Private Networks (VPNs). Our study shows that our techniques effectively extract correlated configuration changes, within and across individual routers, and shed light on the prevalence and causes of system-wide and intertwined change operations. A deeper understanding of correlated changes has potential applications in the design of an auditing system that can help operators proactively detect errors during change management. To demonstrate this, we conduct an initial study analyzing the prevalence and causes of anomalies in system-wide changes.

1 Introduction

One of the most challenging tasks for IP network operators involves making and managing changes to router configurations that are needed to reflect changes in network designs, or as a response to address network problems. Configuration changes are often *system-wide* (involve most routers in a network) and *intertwined* (require modifications to multiple parts of a configuration file or localized groups of routers). Once configuration changes are made, these dependencies make it difficult for an operator to verify that the changes executed conform to his intent. Even worse, a small but incorrectly applied change can have serious impacts such as Service Level Agreement (SLA) violations for providers, and service disruptions for customer enterprises [1, 2, 3]. Yet, the goal of correctly making and effectively managing configuration changes remains daunting for operators, considering the large size and geographical span of networks, the myriad of configuration options, and the variety of routers from different vendors.

Existing tools (e.g., [4, 5, 6]) for automated change management are inadequate when coping with dependent changes for two reasons. First, typical tools are geared towards

S.B. Moon et al. (Eds.): PAM 2009, LNCS 5448, pp. 111–121, 2009.

managing one router at a time. Second, changes are tracked using device and vendor-specific low-level languages, and deal with myriads of details such as line card settings and routing parameters. Without a network-wide view of what changed and how changes were related, it is difficult for an operator to gauge the network state, verify changes were executed correctly, and know where to look for sources of potential or existing problems.

This paper introduces a methodology that extracts network-wide correlations of configuration changes (a group of changes that consistently occur together) and their high-level intent from low-level router configuration files. To do this, our methodology (i) identifies syntactic configuration blocks that changed by abstracting away low-level details, (ii) applies data mining techniques to expose correlated changes, and (iii) highlights changes of interest via operator feedback. We use router configuration files since they are considered by the operational community to be the most accurate source of records of changes. Distinct from prior works [7, 8, 9, 3] based on static configuration snapshots, we focus on developing longitudinal views of *changes across time*.

One distinguishing feature of our methodology is the use of data mining techniques. From our experience, operator knowledge tends to be incomplete. In particular, given the multitude of different configuration options, networks managed, and operator teams involved, it is nearly impossible for an operator to explicitly list all changes of interest up-front, not to mention that there may be hidden changes an operator may be unaware of. Employing data mining techniques enables automatic discovery of an initial set of correlated changes that are potentially important, without operator support. Yet, correlation does not always imply meaningful relationship. To address this, we corroborate the uncovered correlations with operators and only highlight meaningful ones.

Employing our methodology, we conduct a longitudinal study of changes made in enterprise VPNs, one of the most dynamic and demand-driven services that ISPs provide to customer enterprises today. We analyze a collection of daily snapshots of configurations files pulled from routers in operational networks. Our datasets include 5 enterprise VPNs, each consisting of a few hundred routers, over a 11-month period. Our analysis confirms the value and effectiveness of our methodology, and conveys important insights on change behavior in these networks. Finally, we conduct an initial study analyzing anomalies in system-wide changes as a demonstration of how our methodology can provide insights that help operators detect errors in change operations.

2 Methodology

2.1 Exploiting Syntactic Structure for Change Characterization

Existing router configuration languages are low-level and vendor-specific. The key issues we face are determining what the right abstraction is to associate a configuration change with and how to easily obtain the abstracted configuration for different vendor languages. We could abstract changes based on coarse semantic meanings (e.g., related to Wide Area Network versus Local Area Network), or low-level attributes (e.g., bandwidth 10 versus 100). However, both choices require detailed domain knowledge for the configuration language under study and are not feasible given the complexity and heterogeneity of today's configuration languages. We therefore choose to abstract changes based on the syntactic structure of configuration languages.

```
1 interface Ethernet0/0
2  ip address ......
3 !
4 interface ATM1/0
5  ip address ......
6 !
7 access-list 2 permit host 10.1.1.1
8 access-list 2 permit host 10.1.2.2
9 access-list 3 permit host 10.1.3.3
```

Fig. 1. A Cisco router configuration

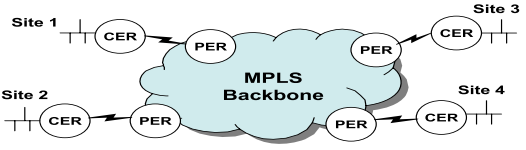

Fig. 2. An enterprise VPN with four sites

To demonstrate this in the Cisco IOS language, consider a Cisco router configuration file in Figure 1. The configuration may be viewed as consisting of multiple *blocks*. Each block comprises a first line that is not indented, and a series of indented lines associated with the block. For example, lines 1-3 constitute a block (interface Ethernet0/0). We also consider commands that lack a similar nested structure but share a common prefix, e.g., lines 7-8, to belong to a single block, in this case (access-list 2). For each non-indented line, we term the initial consecutive series of IOS keywords as a *superblock prefix*. Blocks with the same superblock prefix belong to the same *superblock*. For example, Lines 1-6 and 7-9 belong to superblock (interface) and (access-list), respectively. In this work, we focus on understanding configuration changes at the superblock level. Finally, our initial study of configuration languages from other vendors, such as Juniper and Alcatel, suggests that they bear similar block structures and are amenable to a similar approach.

2.2 Algorithm for Extracting Correlated Changes

To systematically extract correlated changes, we employ the *Apriori* algorithm [10], a powerful data mining technique for association rule induction. *Apriori* is typically used for *market basket analysis*, which aims at finding regularities in the shopping behavior of customers. It expresses an association between *items* within a *transaction*. An *association rule* is in the form "Given a transaction, if a set of items (or itemset) X occurs, itemset Y also occurs," or X→Y.

The standard measures to assess goodness of a rule are its *support* and *confidence*. Let T be the set of all transactions. The support of an itemset X, S(X), is the percentage of transactions in T in which all items in X occur together. The confidence of a rule X→Y, C(X→Y), is defined to be the percentage of times that the occurrence of X implies that all items in X and Y occur together, i.e., S(X,Y)/S(X)*100%. Using these measures, an itemset X occurs frequently if S(X) is high. A rule X→Y with a high confidence makes a good prediction about the occurrence of Y given that X occurs. If the average of C(X→Y) and C(Y→X) is high, itemsets X and Y are strongly predictive with respect to each other, i.e., they consistently occur together. We extend this idea to find *clustered* itemsets, where in each itemset R with n items, the average of C(Subset of n-1 items from R→ n_{th} item) exceeds a threshold t_c.

We employ *Apriori* in the context of finding which routers or superblocks tend to change together in a network. For example, to find correlated routers, we define the set of transactions to contain days on which some router changed and the set of items in a transaction to be the routers that changed on a day. Similarly, to find correlated superblocks, we define a transaction to correspond to each instance of a router change

on a given day, which we refer to ⟨router,day⟩, and the set of items to be the superblocks that changed in a router on that day. In our context, an association rule would be "If routers x and y change together in a day, router z always changes on that day," or "If some access-list (ACL) changes in a router, 50% of the time, some interface of that router changes, too."

3 Characterizing Changes in Enterprise VPNs

Fig. 2 shows a customer enterprise VPN spanning multiple sites over an Multi Protocol Label Switching (MPLS) provider backbone. Each site typically has a customer edge router (CER) connected via a WAN link to a provider edge router (PER). End-to-end Class of Service (CoS) is provisioned by marking packets and treating them differently according to their markings, on the CER-PER-backbone-PER-CER path. The dynamic and heterogeneous nature of changes to CERs [11] makes them the focus of our study.

Changes to CERs may be initiated by the customer and the provider. Events such as changes to passwords may be initiated by the provider as bulk updates in a VPN, or ISP-wide updates across multiple VPNs. Changes driven by customers might be planned, e.g., an interface or link upgrade. Unplanned changes might relate to troubleshooting customer complaints. A high-level change demand may involve a large number of CERs and operator teams.

Some changes are primarily controlled and maintained by the provider. This includes changes to passwords, packet and route filters, and management services like network time protocol (ntp) servers. Other changes, such as changes to CoS and routing designs, may be initiated by both the customer and the provider.

3.1 Datasets

Our data includes 11 months of daily archives of CER configuration files from 5 operational enterprise VPNs. We study longitudinal snapshots of configuration files for two reasons. First, configuration files are considered by network operators to be the ultimate and most accurate source of records of changes. Second, router configuration files are widely available in any network, ensuring our methodology is generally applicable. One data source we did not use is logs that show the sequence of low-level configuration commands executed by the operators. Such logs may enable us to directly reason about operator actions. However, information in these logs may be incomplete - it is possible for operators to bypass the logging system, particularly when bulk changes are involved.

Table 1 summarizes our datasets. All CERs are Cisco routers, and all 5 VPNs, E1-E5, are managed by the same provider. The total size of our data is 32GB. The networks were selected to cover a range of different characteristics in terms of size, geographic span, and growth. E3 and E5 had routers all in one country. At the other extreme, E2 and E4 spanned more than 30 countries and 5 continents. *Net Growth* and *Birth Rate* respectively represent the net change in network size (in terms of the number of CERs) and the total number of new CERs added over the 11-month period. Overall, E5 was the most stable due to its low Net Growth and Birth Rate. Configuration file sizes are also diverse within each VPN because different routers may have different roles (e.g., hub versus spoke), and different sites can have different local policies (e.g., a site hosting critical web services requires additional CoS and security configurations).

Table 1. Enterprise VPN data set. The number of CERs per network is between 150-420.

VPN	City	Ctry	Ctinent	Net Growth	Birth Rate	Config Size(# lines)		
						Min	Med	Max
E1	158	2	1	-1.47%	8.21%	408	1033	1487
E2	100	31	5	5.96%	16.56%	320	652	1175
E3	269	1	1	25.2%	25.2%	551	633	1622
E4	162	36	5	7.11%	25.26%	426	767	1475
E5	346	1	1	-2.85%	1.66%	436	489	1104

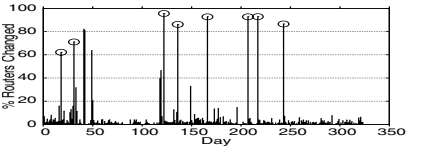

Fig. 3. Percentage of routers changed over time for E2

3.2 Macroscopic Overview of Configuration Changes

We now present key high-level characteristics of changes that we discovered across and within individual CERs.

• *Changes are predominantly local.* Across all networks, in 90% of the days, only 10% or fewer CERs were changed, and in 3% of the days the impact was widespread - with more than 80% of the CERs impacted. Fig. 3 shows the time series of the percentage of CERs changed for E2. Some system-wide changes (shown by high spikes) covered most of the CERs, and were often followed by changes to the remaining routers in the subsequent days. In addition, we found that many large-scale changes were correlated (circled spikes in Fig. 3) across networks. This is consistent with the fact that the provider may schedule ISP-wide changes to several VPNs in the same maintenance window.

• *Some routers are significantly more volatile.* For all 5 VPNs, almost all CERs (98.7%-100%) changed at least once, but the frequency was quite skewed. More than 90% of routers had changes on 3-6% of the days while a small fraction (around 2%) of routers were significantly more volatile, changing on 10%-35% of the days. We found that these volatile CERs usually corresponded to hub routers responsible for switching inter-spoke traffic. Therefore, changes to their configurations were often triggered by changes to spoke routers, e.g., adding an ISDN username/password for a new spoke site.

• *Most changes impact few superblocks.* Most configuration changes were limited to a small number of lines or localized to few superblocks. However, few changes impacted many lines or multiple parts of a configuration file. For example in E1, 58% of $\langle$router,day$\rangle$ instances had $\leq$10 lines changed while only 15% of them had $>$100 lines changed. In addition, 76% of $\langle$router,day$\rangle$ instances had changes to 2 or fewer superblocks.

• *Some superblocks consistently change more frequently.* We define the frequency of a superblock change as the percentage of the days that particular superblock changes in some CER per VPN. In all VPNs, superblock (interface) changed the most frequently while superblock (access-list) was among the top 10 frequent changes. Frequencies of changes to other superblocks were more varied. Other notably volatile superblocks were CoS-related: (policy-map) & (class-map), and routing-related: (router bgp).

4 Correlation Analysis of Changes

To demonstrate the value of our methodology in extracting correlated changes, we perform an in-depth analysis of the 5 VPN datasets. We then highlight particularly interesting correlated changes (i.e., system-wide and intertwined changes) that we corroborated with the operators managing these networks.

4.1 Correlated Changes across Routers in a Network

System-wide Changes. We consider days where a large fraction of routers in the enterprise changed. As shown in §3.2, system-wide changes could spread over a small number of days. Therefore, we consider a global event (or system-wide change) to be a window of w consecutive days where more than $f\%$ of all routers were modified. We pick $w = 2$ and $f = 80$ since Fig. 3 showed that most large-scale changes (high spikes) impacted $\geq 80\%$ of the CERs and were followed by few small-scale changes (short spikes). In the end, we identified a total of 51 global events across all 5 VPNs. This heuristic may miss events which impact all routers but are spread out over a prolonged period of time, and we discuss detecting some such changes in §4.2.

Next, to further understand the nature of global events, we apply *Apriori* (see §2.2) to extract superblocks that consistently changed together in CERs involved in each event. We observe that system-wide changes are typically homogeneous - in each global event, at least 80% of the CERs showed some change in one particular superblock. Among the 51 global events, only 8 events were related to CoS and the remaining were related to management and security operations. CoS-related changes were changes to ACL rules that specify flow memberships of traffic classes. All security changes were changes to ACLs and passwords to control access for remote telnet sessions and SNMP MIBs. Management changes were related to functions such as specifying when SNMP traps must be triggered, increasing the log buffer size, or setting the time zone.

Router Clusters. We consider correlations across small groups of routers that changed for each VPN. Note that a single global event impacts most routers and has the potential to skew this analysis. We therefore filter out days that were a part of some global event. We use *Apriori* to generate clustered router groups with t_c=80 for each VPN. Overall, 1, 4, 26, and 2 clustered router groups are reported for E1, E2, E4, and E5, with an average size of 2, 3, 9, and 3 CERs, respectively. A predominant trend is that the identified clusters show strong geographical proximity, with routers belonging to the same country, or continent. E2 and E4 have more clusters with a larger size on average because they are geographically widespread. Further discussions with the operators revealed a number of reasons for such regional clustering of changes. From a provider perspective, certain changes are administered by operators in different regions, while others are applied centrally. From the perspective of a customer enterprise, VPN sites in different regions may have different local needs, e.g., multiple hub-sites may be configured similarly in a primary-backup setup for resiliency reasons.

Table 2. Most frequently changed superblock groups within a router in E5

N-tuple	support (%)	conf.(%) of n-1 tuple
(access-list)	42.2	NA
(interface)	39.2	NA
(router bgp)	38.9	NA
(ip route)	38.9	NA
(access-list) (router bgp)	38.7	91.7-99.4
(access-list) (route-map)	38.6	91.3-100
(access-list) (ip route)	38.4	91.1-98.9
(access-list) (router bgp) (route-map)	38.2	98.9-100
(access-list) (ip route) (router bgp)	38.2	98.9-100
(access-list) (ip route) (router bgp) (route-map)	38.0	99.4-100
(username)	28.6	NA
(interface) (username)	28.4	72.5-99.6

Table 3. 3 most frequent superblock clusters for each VPN. C:CoS, M:Management, R:Routing.

Ent	Cat.	Superblock Group
E1	C	(interface) (policy-map) (class-map)
		(interface) (ip access-list extended) (policy-map) (class-map)
	R	(router bgp) (route-map) (ip access-list standard)
E2	C	(interface) (policy-map) (class-map)
		(interface) (access-list) (policy-map)
	M	(ntp server) (logging) (snmp-server host)
E3	C	(interface) (class-map) (policy-map)
	R	(interface) (access-list) (router ospf) (ip host)
	M	(logging) (ntp server) (snmp-server host)
E4	C	(interface) (policy-map) (class-map)
		(interface) (access-list) (policy-map) (class-map)
	M	(snmp-server host) (ntp server) (logging)
E5	C	(access-list) (interface) (policy-map)
	R	(access-list) (ip route) (router bgp) (route-map)
	M	(interface) (username)

4.2 Correlated Changes across Superblocks in a Router

Intertwined changes performed by operators may involve changing multiple superblocks. Our goal is to identify correlations across superblocks that consistently change together. To avoid skewing our results, we filter out days involved in global events.

Table 2 shows the groups of superblock(s) that change together most frequently for E5, sorted in decreasing order of support. Superblock (access-list) changes occur in 42.2% of the transactions (i.e., all ⟨router,day⟩ instances), while superblock (router bgp) changes occur in 38.9% of the transactions. Further, the pair of superblocks (access-list) and (router bgp) change together in 38.7% of the transactions. However, superblocks that change frequently individually need not change frequently together. For example, both (access-list) and (interface) individually change in over 40% of the transactions, but they change together in only 2.8% of the transactions (not shown). The right-most column summarizes the range of confidence values C(Subset of n-1 superblocks$\rightarrow n_{th}$ superblock) for all possible subsets of size n-1. For example, superblocks (access-list), (ip route), and (router bgp) occur together in 38.2% of the transactions, and if any two of the superblocks change, the percentage of times that the third superblock changes ranges from 98.9-100% depending on which two superblocks occur.

We now illustrate the types of correlated changes our methodology can identify.

Staggered System-wide Changes. A striking observation from Table 2 is that the group with superblocks (access-list), (ip route), (router bgp), and (route-map) occurs in 38% of the transactions. Further, for any 3-tuple combination of these superblocks, the confidence range is very high ($>$95%). Further investigation with the operator revealed that E5 experienced a change in its network design during the measurement period. The design moved away from using the provider's ISDN backup solution to a solution that points all traffic back to the customer environment in the event of the primary link failing, since the customer had added an additional service provider. The design change consists of modifications to the BGP configuration, additions of access-lists, route-maps and weighted static routes. These changes were introduced over a period of 2 months, configuring

roughly 20-30 sites every 2-3 days. The system-wide change was spread over time to reduce the risk of adversely impacting the primary network traffic. Another superblock group, (interface) and (username), also has a relatively high support of 28.4, with a high confidence range. This turned out to be related to the second part of the same overall design change - removal of the existing ISDN backup solution, which involved deletions of usernames and logical ISDN interfaces, and modifications of physical interfaces referring to the removed logical interface. The slightly lower support for this group is because not all sites of E5 had an ISDN backup solution. Interestingly, these two groups of staggered design changes were performed by independent design teams, and the operators found our methodology useful in confirming these changes occurred as intended.

Frequently Occuring Superblock Clusters. Table 3 summarizes the 3 most frequent superblock clusters for each VPN. For enterprise E5, the 4-tuple corresponding to BGP policy addition, and the 2-tuple corresponding to ISDN backup removal are shown. For each superblock cluster, we assign a category of operation associated with the group. For all VPNs, we find that most intertwined superblock changes are centered around CoS (e.g., provisioning a new class of traffic) and routing (e.g., installing new backup routes), confirming the central role of CoS operations in all the VPNs we consider.

Syntactically Unrelated Meaningful Correlations. Table 3 shows that E2-E4 has a strong correlation in management operations related to ntp, logging and snmp-server. By merely looking at the configuration commands, it would not be clear how these superblocks are related since they do not directly refer to one another. Yet, they turn out to form a semantically meaningful correlation that reflects the periodic server update routines used in those VPNs. This type of correlation involves changes to syntactically unrelated parts of a configuration file. A parser incorporating knowledge of the configuration language itself would be incapable of extracting such correlation. This finding illustrates the potential benefits of our methodology in extracting syntactically unrelated, but semantically meaningful correlations.

5 Application – Finding Anomalies in System-Wide Changes

Knowledge of correlated changes has potential applications in detecting errors in the change management process. In this section, we conduct an initial study focusing on anomalies in system-wide changes.

A key observation we made when analyzing system-wide changes (§4.1) is that some system-wide changes impacted most, but not all, CERs. An auditing tool can leverage such insight and proactively look further for CERs missing an initial bulk update, which we call *outliers*. We analyze their prevalence and further investigate their causes based on operator responses.

We call outliers that never received the missed global update *persistent outliers*. Among the remaining outliers that eventually saw the missed global update, 80% of which received the update within 8 days. The operator indicated that these "short-lived" outliers' initial misses were due to network congestion or routers being overloaded, and were shortly fixed later by their auditing scripts. Therefore, we exclude them from our analysis and call the rest of outliers *delayed outliers*. Table 4 summarizes the outliers.

Table 4. Summary of global outliers detected. Numbers in parentheses denote the number of unique outlier routers and the number of events in which some indicated outlier occurred.

Ent	Total	Persistent Outliers			Delayed Outliers
		errors	non-errors	unknown	
E1	172(38,8)	0	134(26,8)	3(3,2)	35(12,6)
E2	24(15,6)	11(6,3)	7(7,1)	5(5,3)	1(2,1)
E3	9(8,2)	0	2(2,2)	7(7,1)	0
E4	91(85,7)	0	81(78,2)	6(6,2)	4(4,3)
E5	16(3,6)	0	10(10,1)	0	6(3,4)

Note that a CER may appear as an outlier multiple times if it missed more than one global update. We present some interesting causes for these outliers below.

• **Persistent Outliers:** We classified persistent outliers into *errors* and *non-errors*. In a few cases, we were not able to determine the causes, and we classified them as *unknown*. (i) *Errors*: These outliers were confirmed by the operators as needing fixes. We found 11 such outliers, all in E2. They corresponded to missed management updates, e.g., increasing size of logging buffers and setting timeout for management sessions. The operators indicated that although these errors were not critical to essential operations of the VPNs, it is important that all operators are aware of the existence of these errors in order to evaluate their potential impact, and take remedial actions if needed.

(ii) *Non-Errors*: These outliers were either confirmed or strongly suspected by the operators as genuinely not needing the update. They constitute the majority of outliers detected in E1, E4, and E5. The 134 cases in E1 involve only 26 routers, all related to CoS design. For example, a small fraction of CERs are located in a different country from all other CERs, and they use a different CoS design and have different update patterns. Non-errors in other networks were management-related. For example, low-end routers did not get the complete set of management ACL rules to reduce processing overhead. In addition, updates that increase certain parameter values (e.g., logging buffer size) above a threshold did not reach routers which already had them above the threshold.

• **Delayed Outliers:** Table 4 shows that E1 had the most delayed outliers, but on only 12 CERs. These routers used a newer Cisco style of CoS configurations which required manual updates because they were not amenable to bulk updates through older management tools. In addition, while E2-E5 allow fixes to be made on-demand, E1 had a more stringent update process in that changes can be made only in pre-scheduled time windows. These two factors explain a large number of delayed outliers in E1. For E2-E5, one major cause of delayed outliers was that the misconfiguration of management ACLs inadvertently blocked global updates.

6 Related Work

To our knowledge, the only other work that has analyzed dynamic operational tasks of real networks is [12]. While we share similar high-level objectives, [12] tries to identify groups of syntactically related commands and builds models to describe the series of actions needed for an operator to perform a given task on a router interface. In contrast, we focus on automatically extracting correlated changes within and across

routers using data mining techniques. Minerals [7] also uses association rule mining to analyze configurations. However, they focus on detecting misconfiguration using static configuration snapshots.

Several works have sought to automate top-down generation of low-level configurations [11, 13] and typically focus on greenfield deployments. Others [3, 8, 9] have looked at detailed modeling and detection of errors in static configuration snapshots. Many of them are device-specific, and do not help operators understand and explain the semantics behind changes.

7 Summary

In this paper, we have presented a methodology to extract network-wide correlations of configuration changes from longitudinal snapshots of router configuration files. Our study of five operational enterprise VPNs over an 11-month period confirms the value and effectiveness of our methodology, and conveys important insights on the change behavior in these networks.

Our results show that while most changes affect individual routers, system-wide changes do occur, and primarily relate to management and security operations. In addition, correlations exist across groups of routers located in geographic proximity to each other. When correlations across superblocks were considered, most of these corresponded to changes to the CoS or routing design. Interestingly, one of the networks exhibited a markedly higher frequency of superblock groups - further analysis indicated this corresponded to a system-wide design change that was staggered over multiple days. Also of interest, our analysis revealed meaningful yet syntactically unrelated correlations, arising due to management processes employed in the networks.

While our findings are specific to the networks we analyzed, the methodology itself is generally applicable to all networks. A potential application is in tools that can provide operators with network-wide summaries of changes applied to their networks. Extracting correlations also has potential applications in the design of change auditing systems, that can alert the operator to violations of these correlations during a configuration change. We illustrate the potential of this direction by presenting an initial study of anomalies in system-wide changes in §5.

Much of this study has been performed with active involvement of operators, who have expressed great interest in the methodologies and findings. We are currently in the process of developing tools based on our methodologies for summarizing and auditing configuration changes.

References

[1] Mahajan, R., Wetherall, D., Anderson, T.: Understanding BGP misconfiguration. In: SIGCOMM (2002)
[2] Kerravala, Z.: Configuration management delivers business resiliency. The Yankee Group (2002)
[3] Narain, S.: Network configuration management via model finding. In: LISA (2005)
[4] Cisco IP solution center, `http://www.cisco.com/en/US/products/sw/netmgtsw/ps4748/index.html`

[5] Intelliden, http://www.intelliden.com/
[6] Opsware, http://www.opsware.com/
[7] Le, F., Lee, S., Wong, T., Kim, H.S., Newcomb, D.: Minerals: using data mining to detect router misconfigurations. In: MineNet (2006)
[8] Feamster, N., Balakrishnan, H.: Detecting BGP configuration faults with static analysis. In: NSDI (2005)
[9] Feldmann, A., Rexford, J.: IP network configuration for intradomain traffic engineering. IEEE Network Magazine (2001)
[10] Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: VLDB (1994)
[11] Enck, W., McDaniel, P., Sen, S., Sebos, P., Spoerel, S., Greenberg, A., Rao, S., Aiello, W.: Configuration management at massive scale: System design and experience. In: USENIX (2007)
[12] Chen, X., Mao, Z.M., van der Merwe, K.: Towards automated network management: Network operations using dynamic views. In: INM (2007)
[13] Gottlieb, J., Greenberg, A., Rexford, J., Wang, J.: Automated provisioning of BGP customers. IEEE Network Magazine (2003)