Computer Networks 65 (2014) 84–98

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Survey Paper

Network monitoring: Present and future

Sihyung Lee [a,*], Kyriaki Levanti [b], Hyong S. Kim [c]

[a] Seoul Women's University, 621 Hwarangro, Nowon-Gu, Seoul 139-774, South Korea
[b] Amazon.com, Inc., 410 Terry Avenue North, Seattle, WA 98109, USA
[c] Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA

ARTICLE INFO

Article history:
Received 22 February 2013
Received in revised form 4 March 2014
Accepted 17 March 2014
Available online 24 March 2014

Keywords:
Network configuration
Design
Management
Measurements
Monitoring

ABSTRACT

Network monitoring guides network operators in understanding the current behavior of a network. Therefore, accurate and efficient monitoring is vital to ensure that the network operates according to the intended behavior and then to troubleshoot any deviations. However, the current practice of network-monitoring largely depends on manual operations, and thus enterprises spend a significant portion of their budgets on the workforce that monitor their networks. We analyze present network-monitoring technologies, identify open problems, and suggest future directions. In particular, our findings are based on two different analyses. The first analysis assesses how well present technologies integrate with the entire cycle of network-management operations: design, deployment, and monitoring. Network operators first *design* network configurations, given a set of requirements, then they *deploy* the new design, and finally they verify it by continuously *monitoring* the network's behavior. One of our observations is that the efficiency of this cycle can be greatly improved by automated deployment of pre-designed configurations, in response to changes in monitored network behavior. Our second analysis focuses on network-monitoring technologies and group issues in these technologies into five categories. Such grouping leads to the identification of major problem groups in network monitoring, e.g., efficient management of increasing amounts of measurements for storage, analysis, and presentation. We argue that continuous effort is needed in improving network-monitoring since the presented problems will become even more serious in the future, as networks grow in size and carry more data.

1. Introduction

Monitoring a network is crucial to network-management operations, and it is used for many critical tasks. A major function of network monitoring is early identification of trends and patterns in both network traffic and devices. According to these measurements, network operators understand the current state of a network and then reconfigure this network such that the observed state can be improved. For example, the operators find a dramatic increase of P2P traffic, which begins to drop most of the other packets. In response to this problem, operators initiate rate-limiting of P2P traffic. The operators may also find that vulnerability in database-servers allows illegitimate access to sensitive information, and then they begin to apply patches to the database servers. Late detection of such incidents can lead to prolonged disruption to services and financial losses up to millions of dollars [1].

Due to the significance of network-monitoring operations, an extensive amount of work was done to advance these operations. However, network operators still spend the most of their time monitoring and troubleshooting

[*] Corresponding author. Tel.: +82 2 970 5608; fax: +82 2 970 5974.
E-mail addresses: sihyunglee@swu.ac.kr (S. Lee), kiki.levanti@gmail.com (K. Levanti), kim@ece.cmu.edu (H.S. Kim).

http://dx.doi.org/10.1016/j.comnet.2014.03.007
1389-1286/© 2014 Elsevier B.V. All rights reserved.

problems in their networks [2]. Network outages continue to occur and prevent access to networks for several hours. For example, more than three hours of outages were reported in both Amazon [3] and YouTube [4] networks. As a result, enterprise networks spend an increasing amount of their IT budget in network monitoring, rather than to add new value-adding services and equipment [5].

Considering the significance and complexity of network monitoring, we identify challenges in network monitoring, summarize existing solutions, and suggest future directions for dealing with the challenges. This was performed by analyzing existing works published in 8 journals and the proceedings of 13 conferences[1] for the past 15 years (i.e. from January 1998 till December 2013), and by then selecting a set of significant challenges. This paper can be used in several different ways. It can help researchers better understand missing points in current practices of network monitoring and thus conceive new ideas that can improve the status quo. This paper can also be used to study a wide range of monitoring operations and their relationship with other network-management operations. We summarize the suggested guidelines in Table 1.

The scope of this paper is the management of one administrative domain in the Internet, and it covers different types of networks, such as ISPs, enterprise networks, and campus networks. Potential readers of this paper include researchers, network operators, as well as students.

Related work: A few surveys on network management exist, but their main focus is different from this paper. Some of these surveys are dedicated to sub-topics discussed in this paper [6,7]. Other surveys present several network-management issues in general [8,9]. In contrast, this paper is aimed at presenting a holistic view of network-monitoring operations. To this end, we examine both the internals of monitoring operations (e.g., efficiency improvement in packet sampling) and monitoring in relationship with other operations (e.g., automation of the interaction between monitoring and configuration design). [6] introduces several monitoring functions that analyze network traffic, such as traffic classification and application discovery. [7] presents an overview of one particular function of network monitoring, fault localization. The functions presented in [6,7] are components of the analysis layer, one of the five layers of monitoring, as we discuss in Section 3.5. [8,9] summarize several network-management issues, some of which are related to network monitoring (e.g., efficient storage of packet measurements in the face of large traffic volume). These issues are also discussed in this paper. To summarize, this paper focuses on network-monitoring operations, covering a wide range of problems in network monitoring.

The following sections are designed to analyze monitoring from a few different angles, in order to identify diverse areas that can be improved in monitoring. In Section 2, we present the position of monitoring in relationship with other network-management operations. We then utilize this relationship and suggest guidelines that improve monitoring. Sections 3 and 4 delve into monitoring operations and classify these operations into five different categories according to their functions. In particular, Section 3 describes challenges for each of the five categories, and Section 4 presents the challenges that are shared by multiple categories. The two sections also highlight existing solutions and future research directions. Finally, we summarize the proposed research directions and conclude in Section 5.

2. Monitoring within the big picture of network management

In Section 2.1, we first position monitoring in the entire cycle of network-management operations. This positioning of monitoring is then used in Section 2.2 for analyzing interactions between monitoring and other network-management operations. According to this analysis, we suggest guidelines for improving the operational cycle as a whole. The positioning of monitoring in Section 2.1 can also serve as background information on network monitoring.

2.1. Position of monitoring in the operational cycle of network management

To better understand the limitations of current network-management practices and to identify areas of improvement, we position monitoring within a sequence of operations that are performed as a network evolves: design, deployment, and monitoring. Fig. 1 depicts the three groups of operations and their interactions. We first describe two types of data sources that are frequently used in the three groups of operations. We then explain the details of the three groups and their interactions.

In managing networks, network operators use two types of data sources: *measurements* and *configurations*. Measurements show a network's current behavior, and they include packets collected at different vantage points as well as dynamic device-specific information, such as CPU load and forwarding table entries in a router. The configuration of a network device is a set of device-specific commands and parameters. These commands and parameters specify the device's intended behavior: how the device should operate, which protocols should be running, and what values the protocol options should take. Configurations also include information about the physical and logical connectivity between the network's devices.

Measurements and configurations are used by the three operational groups in the following ways. The *monitoring* operations collect measurements and analyze them, in order to infer the current behavior of a network. By considering this current behavior, the *design* operations create necessary changes in configuration and infrastructure. These changes help fulfill requirements specific to the network (e.g., evenly distribute traffic over N links).[2] The

[1] The 8 journals include Springer JNSM, Wiley IJNM, ACM CCR, IEEE TNSM/ToN/JSAC/Network/Communications.The 13 conferences include IFIP/IEEE IM/NOMS/CNSM, USENIX LISA/NSDI, ACM SIGCOMM/CoNEXT, IEEE POLICY/INFOCOM/DSN/Globecom/ICC/VizSec.

[2] Goals, requirements, and intended behavior (and consequently, configurations) are the results of network design. The network design takes requirements from applications and maps them to physical infrastructure setup, configurations, and goals on the operations.

Table 1
Summary of guidelines for future network-monitoring systems.

Guidelines	Section
Automation of the operational loop (monitoring → design → deployment)	Section 2.2
Improvement of the design-group operations through configuration evaluation, simplification, and visualization	Section 2.2
Simultaneous improvement of monitoring efficiency, accuracy, and flexibility	Section 4.1
Intelligent storage, analysis, and presentation of vast amounts of measurement data	Section 4.2
Consideration of the characteristics in virtualized infrastructure (data-center networks)	Section 4.3

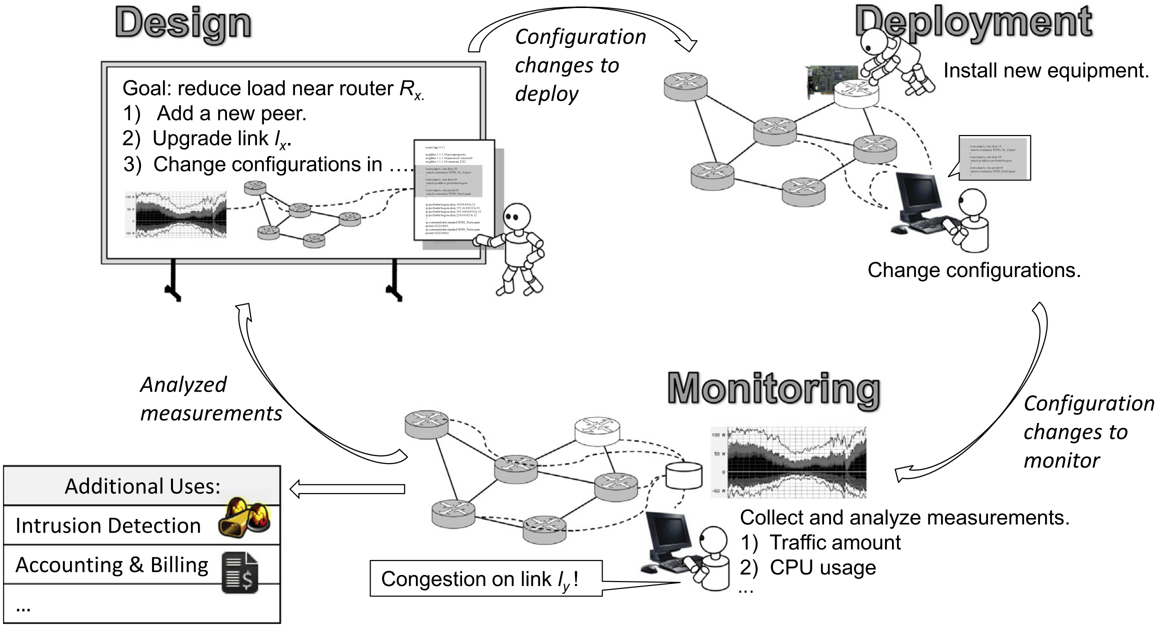


Fig. 1. Classification of network management operations into three groups: (i) *monitoring* of a network's behavior, (ii) *design* of configuration changes according to requirements, and (iii) deployment of configuration and infrastructure changes.

planned changes are deployed in the network by the *deployment* operations. Since most changes are deployed in a live network, minimizing disruptions to the network's operation is the primary goal of the deployment operations. At the same time, the monitoring operations oversee the network's behavior in order to ensure that it operates according to the intended behavior.

The scenario in Fig. 1 illustrates the interactions among the three groups of operations. A traffic surge observed through monitoring results in a plan of changes for the network. The planned changes implement the goal of operating the network without congestion. Network operators deploy the new design and verify the effectiveness of the introduced changes through monitoring. Table 2 present basic network-management functions under the three groups of operations.

We use the term *network model*, when we refer to the current behavior of a network inferred by the monitoring operations. In other words, the monitoring operations build a network model by collecting and analyzing measurements. This network model is extensively used during design operations in two ways. First, the model is used as a reference for designing configurations when new requirements are received. Second, it is used to identify any mismatch between the network's current behavior and the network's intended behavior. In order to deal with such mismatches, network operators re-design the configurations to restore the network's intended behavior. The mismatches occur because of (i) hardware/software flaws in network devices, (ii) faulty configurations (often created by humans), or (iii) configuration assumptions that no longer hold. Configuration assumptions refer to the assumptions over a network's states. These assumptions are typically based on past measurements, and they guide configuration design. For example, a network is configured to evenly distribute flows F1 and F2 between two paths, according to the assumption that the volumes of F1 and F2 do not fluctuate and remain similar. If these volumes significantly deviate from the assumption, the configuration may cause one path to become more heavily utilized than the other. The mismatch can be eliminated by modifying configurations, such that the flows are re-distributed on the two paths, according to the new flow volumes. The mismatches may also reveal threats to the security of a network, thereby triggering alerts in intrusion and anomaly detection systems. According to these alerts, network operators re-design the configurations to mitigate damages and also to prevent further threats (e.g., by adding new firewall rules or intrusion

Table 2
Examples of network-management operations.

Operations group	Network management functions
Monitoring	*Monitor and troubleshoot a network* Measure the network's behavior Identify usage patterns and localize problems in the network Verify the accuracy of configuration changes
Design	*Design configuration changes according to requirements* Design desired behavior according to the requirements Read and understand existing configurations Map the desired behavior into configuration changes
Deployment	*Deploy configuration changes to the network* Safely deliver configuration changes Roll-back to a previous state when changes are not satisfactory Ensure that the software of network devices receives patches and updates

prevention policies). The mismatches can also lead to changes in existing policies and development of new policies. For example, a settlement-free network peer P was found to transfer considerably more data through our network than we transfer data through P. We may then cease the peering relationship and initiate a contract to charge P for the extra transfer. Monitoring helps identify all of the mismatches and deviations by building a network model, as mentioned above. Besides the use in design operations, the network model is also used for accounting and billing purposes, i.e., to track network utilization, so that network users are appropriately charged.

To summarize, the three groups of operations create a cycle of network operations. These operations repeat continuously as mismatches are found, existing requirements change, and new requirements are issued. Within this operational cycle, the monitoring operations play a critical role in the verification of configuration changes and in the identification of deviations from intended behavior.

2.2. Guidelines for improving the operational cycle

In this subsection, we present guidelines for improving the three groups of operations, monitoring, design, and deployment. In particular, we focus on the monitoring operations and also on their interactions with the design and deployment operations.

2.2.1. Automation of interactions between monitoring, design, and deployment

The three groups of operations closely interact with one another: according to the monitored behavior of a network, operators adjust the design of configuration changes, and then these design changes are deployed. However, many proposed network-management solutions address challenges in one of the three groups, rather than focusing on how to efficiently integrate the solutions in the operational cycle. For example, it is often not clear how a measurement analysis in the monitoring operations can be used by the design operations, or how a proposed design can be deployed with minimal disruption.

By automating the interactions between the three groups, operations can be carried out more efficiently. One way to automate the interactions is to use *if-then-else* clauses similar to those in programming languages. Each if-then-else clause specifies (i) a range of network states as conditions and (ii) a set of design options as actions. When a condition is satisfied, the corresponding design option is automatically deployed. This type of automation is called Policy-Based Management (PBM) [10]. For example, path p_A is configured as the primary path for a network flow. If traffic measurements show that p_A has been more heavily used than the secondary path p_B, the automation switches p_A and p_B in order to prefer the less congested path. The traffic measurement corresponds to the monitoring operations, and the following switch between paths corresponds to the design and deployment operations.

Although a number of PBM solutions have been proposed, many network operators manually perform the interactions among the three groups of operations, and they often rely on past experiences and trial-and-error. We present two issues that might have delayed a wide adoption of the PBM solutions. First, each proposed PBM solution requires operators to learn a new, different policy-description language in order to express the if-then-else clauses and to fully utilize the proposed system. Second, the proposed solutions automate particular tasks and need significant customization in order to be applied to different aspects of network operations. For example, one solution efficiently automates the configuration of a primary DNS server and a secondary DNS server in each subnet. However, it may not be straightforward to apply this solution to the configuration of network protocols other than DNS, such as routing. To summarize, a future automation system could be more widely deployed if it addresses the two problems in the existing PBM solutions.

2.2.2. Improved design

Network operators spend a large amount of time monitoring and troubleshooting problems in their networks. More than half of these problems can be traced back to operator errors in configuration design [11]. Therefore, configuration design with fewer errors would save network operators significant time since fewer failures would surface during monitoring. We present three groups of methods that can improve the accuracy of design operations and thus can reduce time taken to monitor a network.

The first method of improving the accuracy of configuration design is to utilize configuration evaluation. Configuration-evaluation systems identify errors in configurations before design changes are deployed, and this process is performed according to different sets of correctness criteria. Feldman and Rexford [12] present a tool that performs basic reference and consistency checks between configuration segments. For example, it checks whether the parameters configured on both ends of a link are the same. The router configuration checker (rcc) [13] detects configuration faults related to the visibility of inter-domain routes. For example, it ensures that the routing-protocol sessions are configured such that each router receives routes from all other routers in the network. One drawback of many configuration-evaluation systems is that they are somewhat limited to identifying a predefined set of inconsistencies and cannot be easily extended to other types of errors. However, using a combination of multiple systems can cover a wide range of errors, and therefore significant effort can be saved in the monitoring operations.

The second method of improving the design operations is configuration simplification, which removes obsolete and unnecessary configuration segments, thereby making the understanding of configurations easier for network operators [14,15]. Compared to configuration evaluation, simplification is proactive in the sense that it removes configuration redundancies that increase maintenance costs and operator mistakes. Simplification is particularly useful to networks where configurations have gone through many changes over time and have thus become extremely complex.

Finally, design operations can also benefit from improved visualization systems. Visualization systems help operators understand existing configurations and the effect of a potential change [16]. By illustrating a combination of network configurations and monitored network states, as well as the way these two data interact, operators can more accurately estimate the effect of a configuration change.

2.3. Summary

In this section, we position network monitoring within the big picture of network-management operations, and highlight the roles of monitoring among the operations. This positioning serves as background material for the following Sections 3 and 4, which focus on network monitoring and provide detailed analysis.

Section 2.1 describes that network-management operations are classified into three components, namely, design, deployment, and monitoring operations. These components tightly interact with each other, requiring extra work to proceed. For example, monitoring leads to the following series of operations: (i) monitoring operations identify a performance problem; (ii) this problem needs to be thoroughly analyzed, and correlated with configurations, to identify potential causes; (iii) according to the found causes, the design operations can modify configurations. Step (ii) is additional work necessary to proceed from step (i), monitoring, to step (iii), design. The series of operations can be greatly simplified if we automate step (ii), as shown in Section 2.2. For this type of automation to be widely adopted, we need to develop a standard language that can describe policies in diverse aspects of network operations.

3. Existing solutions, challenges, and future directions for monitoring operations

In this section, we analyze the group of monitoring operations. First, we present a five-layer sub-classification of monitoring operations (Section 3.1). This classification illustrates the different logical functions in monitoring. We then analyze the functions of each layer with an emphasis on how these functions affect network operations (Sections 3.2–3.6). In the next section, we present two issues that challenge all monitoring operations (Section 4).

3.1. Overview of monitoring operations

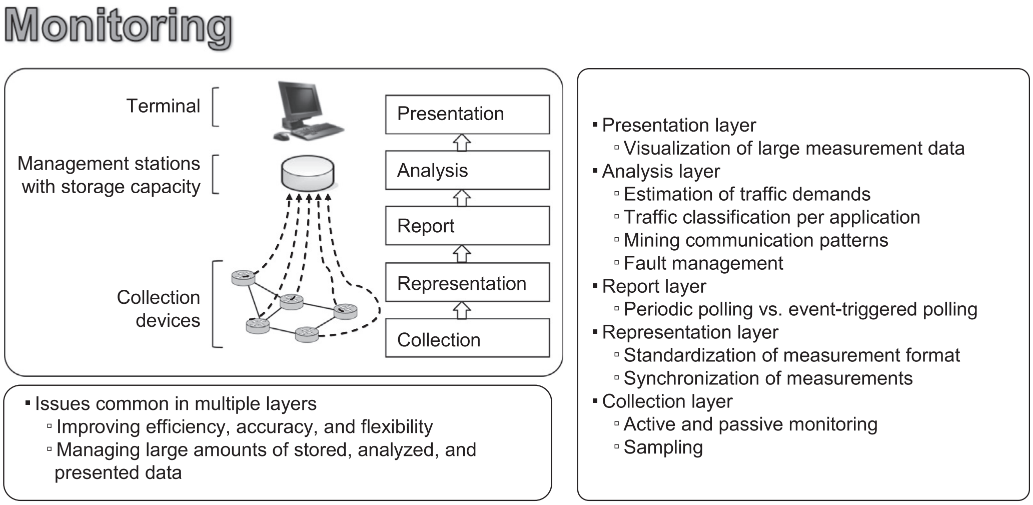
The monitoring operations build a model of the network's current behavior. The network model represents the operational status of the network. It is used for troubleshooting and future planning, and it often triggers a new design. In order to capture the network model, we need the five logical measurement functions illustrated in Fig. 2: collection, representation, report, analysis and presentation. Although many works address problems in more than one layer, this logical separation of the different monitoring functions clarifies their functionality and interactions.

The *collection* layer collects the raw measurement data from the network. This data is then processed and put into a particular format by the *representation* layer. This format is often independent of the management function so that the data can be used by many different management functions. The *report* layer transfers measurement data collected from a number of network devices to a smaller set of management stations where higher-layer functions are typically placed. The *analysis* layer analyzes the measurement data and extracts high-level interpretations of the collected data. Some common analysis functions are traffic classification and fault detection. The *presentation* layer presents measurement data to network operators in different formats, such as visual and/or textual representations.

Note that the five layers are logical layers. The components of the different layers can be implemented either in the same physical device or in different devices. Typically, the top two layers are implemented in one or more management stations and the bottom two layers are implemented by a larger number of collection devices. The collection functionality is usually performed by the network devices that implement fundamental networking functions, such as routing and switching.

A large body of previous works has focused on improving the efficiency of the monitoring process. This is a challenging problem because there are vast amounts of data to collect, analyze and store. The problem becomes more important as link rates increase far above Gbps, IPv6 becomes widely deployed, and the Internet usage increases dramatically [6]. Even when the processing power, bandwidth overhead, and storage capacity no longer pose critical limitations to the monitoring process, the identification of important events among vast amounts of measurement data and the visualization of this data will remain a challenge.

Another large body of previous works on monitoring is related to the analysis layer. The high-level interpretations derived from the measurement data are used to decide whether to alarm the operator, log the reported measurements, or reconfigure the network. Also, most analysis works focus on the network's traffic. Traffic measurement analysis is heavily used to aid a wide range of design operations. Examples of such operations are usage-based billing, capacity planning, and evaluating the network's



- Issues common in multiple layers
  - Improving efficiency, accuracy, and flexibility
  - Managing large amounts of stored, analyzed, and presented data

- Presentation layer
  - Visualization of large measurement data
- Analysis layer
  - Estimation of traffic demands
  - Traffic classification per application
  - Mining communication patterns
  - Fault management
- Report layer
  - Periodic polling vs. event-triggered polling
- Representation layer
  - Standardization of measurement format
  - Synchronization of measurements
- Collection layer
  - Active and passive monitoring
  - Sampling

Fig. 2. Sub-classification of monitoring operations.

capability to support new value-adding services such as VoIP and IPTV. We expect that analysis-layer functions will continue to dominate the monitoring operations as new applications appear and existing applications evolve in time.

3.2. Collection layer

Here we present issues that pertain to the collection of measurement data. Network operators can collect measurement data in an active or passive manner. They can also choose between various methods of sampling the measurement data.

3.2.1. Active vs. passive monitoring

Most of the collection layer functions can be classified into active or passive monitoring. Active monitoring is the monitoring that involves the injection of test traffic, and it usually runs on an end-system. Active monitoring can directly measure what operators want to observe without waiting for a particular event to happen. However, it needs to minimize the impact on normal traffic and ongoing network operation. The size and frequency of the active probing are the two parameters that determine the impact of active monitoring on the network's operation. Also, the test traffic needs to resemble real traffic as much as possible. Otherwise, the observed network behavior may not fully match with the behavior of the network to real traffic. Passive monitoring is the monitoring that depends on network devices to observe and measure the network's behavior. Passive monitoring either runs on dedicated devices or is performed by network devices such as routers and switches. Passive monitoring is non-intrusive and affects the behavior of the network less than active monitoring. Also, it observes the actual behavior of the network. However, operators may need to wait long until they observe a particular event of interest. Given the characteristics of active and passive monitoring, network operators make use of both methods according to what fits best to their monitoring objectives.

Many works on active monitoring estimate the latency, loss rate, capacity, and throughput of links [17,18] or pinpoint bottleneck locations [19–22] monitors how faults are handled by the network by injecting faults in the form of modified packets. UAMA [23] is an active-probe management architecture for collecting and managing information from many different types of active probes. Works that perform passive monitoring include those related to sampling packets and collecting device-specific statistics, such as CPU and memory utilization. Some works use both active and passive monitoring. In [24], each network node inserts a unique packet-marking in proportion to its traffic load, and then edge nodes estimate the network's traffic load by observing these markings.

A few works suggest the addition of functions to existing routers in order to make active monitoring easier and more accurate. [25,26] propose that collection devices insert more information in probe packets. For example, finer-resolution timestamps would allow the fine-grained measurement of queuing behavior and simplify the inference of properties such as link capacity and available bandwidth. [27,28] suggest that routers forward measurement packets with adjustable priority. ICMP packets are treated differently by routers, whereas network operators may want ICMP packets to be treated the same as data packets for some particular measurements. For other measurements, operators may want measurement packets to have the highest priority so that these packets do not experience any queuing delay.

3.2.2. Sampling

Sampling is one way of reducing the overhead on monitoring nodes and performing efficient monitoring. Since unsampled information is lost, most of the works on sampling focus on improving the accuracy of measurements. Some works balance monitoring accuracy and efficiency by dynamically adjusting the sampling ratios. For example, [29,30] increase the sampling rate in low-overhead conditions; when the traffic volume is low or when there is only a small number of flows. [31] uses sequence numbers in packet headers first to infer the number of packets that are not sampled and then to infer the relative size of the different packet flows. RRDTool [32] addresses sample losses and delays by interpolating missing values, so that they appear at constant time intervals.

A handful of previous works propose sampling with the use of a consistent hash function throughout the network. A packet is sampled only when its hash value falls in a particular hash range. This range is set by the operators. In this way, they can precisely control the set of packets to sample according to different monitoring objectives. Trajectory sampling [33,34] suggests the use of the same hash range throughout the network. In this way, the operator can reconstruct the complete trajectory of sampled packets without the need of routing information. The complete trajectory can be used either to estimate traffic demands on different links or to locate forwarding problems such as forwarding loops. cSamp [35] assigns different hash ranges over different collection devices so as to satisfy two goals: minimize redundant sampling at the collection devices, while being able to observe and sample all network flows.

3.3. Representation layer

There are three important issues related to the representation layer. First, the representation of measurements needs to be standardized so that each analysis function does not convert the collected measurements into distinct intermediate forms of representation and that the different analysis functions can use the same collected measurements. SNMP MIB [36], CIM [39], IPFIX [40], NETCONF [37], and YANG [38] provide standards for this purpose. SNMP MIB is the de facto standard for retrieving measurement information from network devices; NETCONF and YANG provides a common and extensible representation of both measurements and configurations, and this representation is based on XML. CIM covers a wider range of objects such as servers, desktops, and operating systems, and the object models are designed according to the object-oriented paradigm; IPFIX defines a standard for flow information collected from network devices. The second issue with the representation layer is that measurements from heterogeneous and distributed collection devices need to be synchronized in time. For example, [41] synchronizes the two measurements at each end of a path in order to estimate the delay of the path. The last issue regarding the representation of measurements is that measurements need to be concise, in order to save storage space and network bandwidth. This is achieved by identifying and removing duplicate measurements and also by combining values to create a derived value.

3.4. Report layer

The reporting of measurements needs to be efficient in terms of the bandwidth consumed for transferring the measurement data from the collection devices to the management stations. One way to reduce the measurement traffic is to suppress the reporting of redundant measurements. [42] suggests the aggregation of measurements with similar characteristics. For example, all flows with packet loss exceeding 1% are reported as an aggregate. [43] batches the same polling requests from different applications into one request. Another way to improve the efficiency of measurement reporting is to use bandwidth-saving encodings of measurements. IPFIX [40] defines such encodings.

Another factor affecting the amount of transferred measurements is the frequency of polling for measurement data. Periodic polling requests for measurements with a predefined frequency, usually in the order of minutes. Event-triggered polling requests for measurements only when a predefined event happens. These two methods present a trade-off between monitoring efficiency and accuracy. Periodic polling is less efficient but it allows the identification of new undefined events. Event-triggered polling is more efficient but the events to be reported should be carefully defined because undefined events go unnoticed.

Measurement data can be distributed and consumed by multiple administrative entities. This is often the case when multiple customers share a set of physical machines in virtualized infrastructure. In such situations, reporting mechanisms need to make sure that certain monitoring information is distributed only to the entities who agreed to share the information. To better support the integrity and confidentiality of transported data, monitoring standards implement authentication and encryption. For example, IPFIX provides standard methods of signing flow records and verifying signatures [121]. It also specifies an encapsulation format that can be used for encryption as well as digital signatures.

3.5. Analysis layer

The operations categorized in this layer extract high-level interpretations of the network's state by analyzing the collected measurement data. Although a variety of analysis functions exist, we present the six most common analysis functions: (1) general-purpose traffic analysis, (2) estimation of traffic demands, (3) traffic classification per application, (4) mining of communication patterns, (5) fault management and (6) automatic updating of network documentation. The results of these analysis functions are extensively used for the design of configuration changes. We explain the details of the six analysis functions in Sections 3.5.1-3.5.6.

3.5.1. General-purpose traffic analysis

Many tools process traffic-volume measurements and produce outputs, which can be later used by other analysis functions. FlowScan [44] computes simple traffic flow statistics such as per-protocol and per-port traffic, and it also plots these statistics. Several open-source tools [45,46] collect, store and perform basic analysis of NetFlow data. These tools perform functions such as flow aggregation, filtering, sorting, and printing in readable formats. Other tools process streams of traffic measurements. Thresh [47] and NetViewer [48] propose methods for defining thresholds for normal values and trigger an alarm when the monitored values surpass the defined thresholds. RTG [49] predicts the future trend of measurements based on the history and a time series forecasting algorithm. It flags measurements that deviate from the predicted trend as potential anomalies.

Aggregation of redundant measurements reduces the amount of data to be analyzed. Typically, two measurements M_1 and M_2 can be aggregated if M_1 and M_2 have similar characteristics and if the following analysis requires only the frequency of such characteristics. For example, flow measurements to the same origin networks can be combined, when we analyze only the number of such flows but not individual records [40]. Measurements can be combined according to various *dimensions*, such as IP address, port number, protocol, and timestamps. Such choice of dimension needs to be consistent across a network, when aggregation occurs at multiple, distributed devices (e.g., aggregation with the same set of dimensions at the same levels of granularity). A consistent set of dimensions allows a central management station to perform network-wide analysis of aggregates [45].

3.5.2. Estimation of traffic demands

Traffic demands represent volumes of traffic that flows between each ingress-egress pair in a network. These traffic volumes can be estimated through the processing of traffic and routing measurements. Traffic demand estimates are used for traffic-engineering, capacity provisioning, billing, and anomaly detection.

The largest group of previous works is aimed at the accurate estimation of traffic demands given limited traffic measurements [50–52]. Measurements are limited because packets are sampled and not all the vantage points in a network are equipped with traffic monitoring. [50] compensates for missing information by using routing information and by then estimating a set of possible egress points for each incoming flow. This model can also estimate traffic demands in “what-if” scenarios, such as link failures or configuration changes. [51] collects additional measurements by deliberately changing link metrics and uses this additional information to estimate the missing data more accurately. [52] reconstructs missing values by leveraging a signal processing technique called compressive sensing. Compressive sensing identifies structures in measurements and then interpolates the missing data according to the identified structures.

Another group of works presents summaries of traffic demands instead of detailed views. [53] presents a summary of traffic-demand estimates across different segments of a network. This summary can be used to quickly locate network segments that experience congestions or failures. As traffic measurements are accumulated over time, other works are aimed at estimating representative traffic demands for a particular period of time. This is done either by averaging [54] or by identifying the most critical traffic-demand snapshots in terms of link utilization [55].

3.5.3. Traffic classification per application

This function classifies traffic into different applications. The three most common uses of traffic classification are: (i) optimizing traffic performance according to the distribution of traffic among applications, (ii) pricing and rate-limiting based on the application, and (iii) filtering attacks and copyrighted contents. The naïve way to perform traffic classification is based on the transport-layer port number, which has been a key discriminator for applications [56]. However, port-number-based classification only identifies applications that use the well-known ports registered with the Internet Assigned Numbers Authority (IANA). Additionally, increasingly more applications dynamically assign ports or use well-known ports of other applications.

The most common and accurate method of traffic classification is the payload-based approach. This approach uses payload signatures in addition to the port numbers [57–59]. Most commercial products are based on this approach. However, payload-based traffic classification has several limitations. First, it is computationally expensive since it needs to inspect the payload of each packet [60]. Second, in order to inspect the packet payload, network operators need to install additional packet capturing tools. This is because payload inspection is not supported by NetFlow and SNMP, the most popular monitoring tools available in the majority of routers. Finally, the inspection of packet payload is becoming increasingly more challenging for both technical and non-technical reasons: packet encryption becomes more prevalent [61] and inspection of packet payload raises privacy concerns.

Another way to classify traffic per application is based on host behavior. This approach uses the communication patterns between hosts along with port numbers [62–64]. For example, the server-client model of Web traffic follows the one-to-many communication pattern, whereas P2P traffic follows the many-to-many pattern. These interactions are observable even with encrypted payload. One limitation of the host-behavior-based classification is that it is not as accurate as the payload-based approach because a single communication pattern may be attributed to multiple applications.

To further increase the accuracy of traffic classification, we can use other packet features, such as protocol, packet size, TCP flags, and flow features; flow features include flow duration, packet inter-arrival times, and the number of packets per flow [56]. Most of the feature-based approaches learn the feature patterns for each class of traffic by using data mining or machine learning techniques [65–68].

We identify two major challenges in traffic classification for the future. First, network traffic patterns change over time as existing applications evolve and new applications appear. For this reason, we need to automate the identification of new application signatures and the updating of existing application signatures. Second, the increased use of encryption challenges the payload-based traffic classification. One solution to this problem is incorporating the behavioral traffic patterns into the payload-based classification.

3.5.4. Mining of communication patterns

Communication patterns show clusters of hosts and their Internet usage patterns both in the spatial and the temporal dimension. They provide knowledge about the actual usage of links, protocols, servers, and applications.

This knowledge gives insights into the root-cause of potential network problems. The identification of communication patterns is possible since the execution of almost all business applications leaves a footprint on the network's traffic. This is because the application execution usually involves access to and communication between networked resources. [69,70] identify clusters of traffic and their characteristics based on data mining techniques. These characteristics include protocol, port numbers, sources and destinations of the traffic, and network bandwidth usage. For example, one cluster represents a P2P file sharing application running among a small set of hosts and consuming 70% of the bandwidth. This P2P cluster may explain a recent dramatic increase in the network's traffic. [71] identifies temporally correlated clusters of flows. For example, one cluster represents the following temporal correlation: if connection A is established, connection B is also likely to be established. This temporal communication pattern may expose heart-beat messages between a host compromised by a Trojan and other hosts in the network.

3.5.5. Fault Management

Fault management includes fault identification and fault localization. Fault identification infers the existence of a network fault, and then the root cause of this fault is identified by fault localization. According to the found cause, the fault is repaired, typically through configuration changes.

Currently, operational troubleshooting may go through three groups of people: (i) the operations staff deals with customer calls and solves a subset of problems that are easy to localize; (ii) the network engineers troubleshoot problems that originate from the machines and services under their supervision; and (iii) the network designers deal with the problems that the previous groups could not handle because these problems require in-depth understanding of network configurations and may also involve interactions among multiple devices. As we move on from one group to the next group, the number of unsolved problems decreases, but the time needed to localize their root causes increases. One method to reduce this time is to automate fault identification, localization, and repair, as one single process, as shown in Section 2.2.

3.5.5.1. Fault identification. A large body of the previous works on fault management focuses on fault localization, while taking for granted that network operators have identified the existence of a fault already. In reality, operators monitor traffic volumes primarily and then become aware of problems that manifest themselves through an increase or decrease in traffic volume. However, the operators cannot easily identify faults that do not have an obvious impact on monitored traffic volumes, and even after examining other types of logs (e.g., syslog [76]), the operators may not notice certain failures. A common example of such a problem is performance degradation. In order to identify this type of fault, many networks still rely on customer calls. This shows that the networks need to analyze more diverse types of measurements to improve fault identification. Active investigation is one way to identify faults that do not trigger visible changes in traffic volume [72,73]. For example, [72] detects unreachable destinations in MPLS-over-IP backbone with the use of end-to-end probing and network topology.

3.5.5.2. Fault localization. Fault localization is performed after a failure is observed. Most of the works in fault localization build graphs that represent the dependencies among different network components. These dependencies show the way failures propagate across a network. Based on the dependency graphs, network operators start from failure symptoms and can trace back to the root cause of the failures. The graph construction can be performed at various granularity levels, from the granularity of machines [74] to the granularity of software processes running in the machines [75]. Fault localization at a finer granularity pinpoints the source of failure more accurately, but it requires more time and efforts to build the dependency graph and to search for the source.

One approach to building dependency graphs is to manually embed the dependencies, based on network topology and protocol specifications. However, manual description of dependencies could be overwhelming for network operators, particularly in large networks, where dependencies continue to change over time. To reduce the difficulty of manual work, Sherlock [74] and NetMedic [75] automatically extracts the dependencies by observing externally-visible behavior of system components (e.g., network traffic generated by a system and application states) and by using timing information of such behavior. For example, if access to a web server occurs in a close proximity in time with access to a DNS server for a significant number of occasions, it is concluded that the two services are related. The analyzed system-behavior can be collected in various ways, such as by capturing packets, monitoring connections per process, and logging network events (e.g., syslog [76]).

In a large network with a high degree of interactions among different components, the corresponding dependency graph is highly connected. In such a complicated graph, a naive searching for root causes can easily lead to a number of false alarms, i.e. potential problem sources that are unrelated to the failure. To reduce false alarms, NICE [77] analyzes failure symptoms and then allows an operator to limit the scope of searching into a more likely area, such as within a router, on a path, in the same link, and in the same OSPF area.

Faults in one network layer can propagate to upper network layers and complicate fault localization process. SCORE [78] and Shrink [79] build a cross-layer model, one that includes propagation of failures from optical links to the IP layer. When a failure is observed in the IP layer, network operators can use this model to trace back to the root causes, i.e. optical link failures.

Finally, unlike other fault-management systems, NetPilot [80] is not aimed at localizing problems. It intends to alleviate the symptoms of a failure until this failure is repaired. Such alleviation is performed through a detour of network traffic around spare bandwidth and redundant devices. Using NetPilot in combination with fault-localization systems reduces disruptions to network services, particularly when an extended amount of time is required to recover from the failure.

3.5.6. Automatic updating of documentation

Network documentation includes a network's topology, policies, and configurations. All of these change frequently as network requirements evolve, traffic patterns change, and faults are fixed. Documentations are extensively used when network operators design changes in network configurations. However, in many networks, documentations are manually updated only occasionally since this task is cumbersome and of low priority. Therefore, documentations are often not fully up-to-date, and this can hinder network-management operations that require knowledge of the network's topology and policies. Existing tools update the documentation of particular aspects, such as topology at different layers [81,82], routing policies [83–85], and versions of software components [86].

3.6. Presentation layer

Network operators find it easier to monitor a network through visual representations, rather than through numerical data. The Multi Router Traffic Grapher (MRTG) [87] and the SNMP Network Analysis and Presentation Package (SNAPP) [88] provide visual representations of SNMP data, such as bandwidth utilization. However, visualization of large measurement data is a challenge – a visualization tool needs to infer which information is important, and this tool also needs to represent the information visually in an intuitive way. Flamingo [89] and the SIFT tool [90] provide intelligent visualization of NetFlow records. In particular, they summarize large amounts of NetFlow measurements by aggregating flows according to the destination or origin network. Users of these systems can select a particular type of aggregation. The users can also drill down into the flow details. More recent tools refine visualization with further analysis. For example, NFlowVis [91] links NetFlow measurements to alerts from intrusion detection systems or public warnings, and it also depicts flow between external hosts and internal communication partners. Such visualization helps reveal communication patterns of both malicious and legitimate network traffic, such as massive distributed attacks and P2P communications. [92] identifies correlated events and presents these events as a group, so that operators can efficiently judge the relevance of alerts. It will become even more important to summarize and present important events among vast amounts of measurement data, as Internet usage increases rapidly and more data are collected.

3.7. Summary

In this section, we classify network-monitoring functions into five different layers. These layers help clearly differentiate elements that comprise monitoring. First, the collection layer pertains to the methods of collecting raw measurement data. These methods need to collect sufficient measurements to build an accurate model of a network's behavior, but such collection should not interfere with normal operations of the network. Second, the representation layer determines the formats in which collected measurements are stored and communicated. These formats need to be understood by both the producers and consumers of the data. Third, the report layer refers to the methods of transferring measured data from collection devices to management stations. These methods make sure that the management stations fetch measured data on time and stay up-to-date. At the same time, such movement of data should not congest network links by suppressing redundant reports. Fourth, the analysis layer includes functions that analyze measurements and then extract high-level interpretations of the network's states. These interpretations can be fed into another analysis or can be used to change current configuration settings. For example, the identification of extreme bandwidth usage by a particular application can lead to the configuration of a new policy that rate-limits the application. Finally, the presentation layer presents the results of analysis in human-friendly formats, such as tables and visualization. Such presentations help network operators quickly estimate relative importance of different measurements and therefore focus their time on higher-priority issues. Although each of the five layers can be improved independently, certain improvements in one layer may trigger changes in the other layers as well. For example, the addition of a new analysis requires the collection of additional measurements. Such collection then leads to the standardization of methods to collect, represent, and report all together. In Section 4, we describe more details about the issues related to multiple layers.

4. Issues across multiple layers of the monitoring operations

After analyzing the previous works on each of the five layers of monitoring, we highlight two major issues that relate to multiple layers[3]: (i) the simultaneous improvement of monitoring efficiency, accuracy and flexibility, and (ii) the storage, analysis and presentation of vast amounts of measurement data. We also elaborate on an emerging issue – (iii) monitoring data-center networks (cloud).

4.1. Improving efficiency, accuracy, and flexibility

One technological trend across the multiple layers in monitoring is satisfying three objectives: efficiency, accuracy and flexibility. Increasing efficiency refers to reducing the measurement overhead, i.e. the CPU overhead on the monitoring devices and the network bandwidth utilization. Improved efficiency also includes reducing the maintenance costs. One way to achieve this is reducing the number of collection devices and management stations for maintenance. Increased accuracy refers to building detailed network models with fewer errors. This results either from collecting more measurement data or from choosing the location and type of measurements in an intelligent way. Increased flexibility refers to the easy modification of monitoring objectives and measured data.

[3] The issues occur in the interaction of more than one layer and thus can be solved more efficiently by considering multiple layers altogether. Among such issues, we chose the two issues, owing to their high frequency and significance in the 8 journals and 13 conference proceedings that we analyzed.

This objective can be achieved through programmable measurements.

The three objectives are inter-dependent and pose trade-offs. The goal is to strike a balance between the three objectives. The following examples illustrate the trade-offs between efficiency, accuracy and flexibility. By collecting data more frequently, we can identify more problems, but the CPU load on the collection devices increases. Also, by deploying more collection devices, we increase the monitoring accuracy but we consume more effort to configure these devices and to ensure that they operate correctly. Then, by collecting more diverse types of data, we increase the flexibility in measurement analysis but we also increase the CPU load on the collection devices.

A few works satisfy more than one objective. cSamp [35] controls the range of packets to sample at each collection device. This methodology increases the accuracy of monitoring by configuring the devices to collect more packets from the flow of interest. It also increases the monitoring efficiency by avoiding redundant sampling. [93] performs delay monitoring for ISPs by selecting metrics and measurement intervals that increase the measurement accuracy without sacrificing efficiency. In detail, they propose to use high quantiles rather than other popular metrics, such as average, median, maximum, and minimum. These quantiles capture changes more clearly than average and median and are less skewed by outliers than maximum and minimum.

Some previous works focus on the monitoring efficiency. One approach for reducing both the network bandwidth consumption and the computation overhead on the management stations is raising the intelligence of the collection devices [94–96]. Collection devices are assigned some analysis functions. Using these functions, the collection devices only report the results of the analysis to the management stations, but they do not report the entire set of raw measurements. In other words, part of the analysis-layer functionality is moved to the collection devices. One problem with this approach is that it requires the collection devices to have adequate processing capabilities and that it reduces the flexibility of analysis at the management stations. HAMSA [97] overcomes this flexibility problem by allowing the programmability of monitoring objectives in collection devices.

The positioning of the collection devices also plays an important role in monitoring efficiency. The goal of the positioning problem is to find a set of locations for the collection devices so that the number of observation points is minimized while a specific monitoring objective is achieved. For example, the goal of monitoring in a network is to keep track of all of the flows in the network. In this case, an optimal positioning reduces redundant measurements by positioning each monitor, such that it collects a distinct set of flows. Finding an optimal solution to the positioning problem is NP-complete. Therefore, most previous works propose approximation algorithms [98–100]. In addition to finding an optimal position of collection devices, the most recent work [101] also adjusts routes to further improve monitoring efficiency.

Flexibility increases by collecting more diverse types of data. [102,103] extend the SNMP MIB to include end-to-end statistics, such as per-flow delay and jitter. ProgME [104] allows more flexible measurements than Cisco NetFlow. For example, traffic amounts can be measured by both destination prefix and type of service, rather than by only one of these two flow attributes. Other works allow operators to program flow measurements so that measurements dynamically adapt to network dynamics. [105,106] identify a heavy hitter flow and then sample more packets from this flow by increasing its sampling rate. Another approach for increasing monitoring flexibility is to provide high-level script languages for the easy programming of new monitoring applications [107,108]. To summarize, flexibility can be increased by allowing monitoring objectives to be programmable. Increased levels of flexibility enable more diverse types of analysis. Flexible monitoring can also improve the monitoring efficiency and accuracy by enabling the change of the measurement-focus to the most important network events.

4.2. Managing large amounts of stored, analyzed and presented data

As measurements are continuously collected, the size of the collected information increases rapidly. This large amount of information needs to be stored, analyzed, and presented to the network operator. Therefore, operators need a monitoring methodology that reduces the size of the accumulated data without reducing the accuracy of the measurement analysis. In addition to the real-time measurement requirements, operators retain a history of old measurements for future analysis. Future analysis involves analysis of long-term trends for network planning and analysis of detailed usage measurements for billing and legal purposes.

The most common way to reduce the amount of data to store and analyze is consolidation. Consolidation gradually decreases the data resolution of past measurements by consolidating the data with the use of different functions (e.g., average, maximum, total). It also discards measurements that are older than a threshold. For example, operators may keep all the data for the past month, average over a week for data older than a month, and discard data older than a year. In order to use consolidation, operators need to define the consolidation function and interval given the accuracy needed by their analysis functions [49].

Aggregation is another common method of reducing the amount of data to be analyzed and presented. For example, the statistics of multiple flows can be aggregated when these flows are destined to the same network. Note that flow measurements can be aggregated according to many different dimensions, such as source IP, destination IP, port number, protocol, and time. Flamingo [89] and SAFEM [109] allow operators to aggregate data along these different dimensions. However, users are fully responsible for choosing the most appropriate dimension for their purposes. It would be helpful if aggregation systems provide users with insights about which dimension would result in the most meaningful aggregation, given collected data and monitoring objectives.

A few recent works take a different approach. Instead of reducing data size, they speed up the analysis of large-scale measurements by leveraging hardware parallelism and distributed-computing platforms. PFQ [110] utilizes multiples cores in CPUs and also multiple hardware queues in NICs (Network Interface Cards). Packets are captured and analyzed through parallel paths from NICs up to applications. Several solutions are inspired by the analysis of big data [111]. Monitoring tasks are allocated to clusters of machines, and this allocation is mostly based on the MapReduce logic and Hadoop [112]. To further improve the timeliness of analysis, BlockMon [113], Apache S4 [114], and Storm [115] apply the stream-processing paradigm to network monitoring. These approaches process measurements as they are produced, and there is no need to store intermediate results.

In any cases, the data explosion problem will be aggravated as link capacity increases and IPv6 becomes widely deployed. The increase in link capacity means more data being transferred. The increase in the distinct number of IP addresses means more distinct flows and communication patterns. Both of these increases will make more challenging the identification of important measurements to focus on.

Finally, the ability to process a large amount data (big data) in real time will facilitate a diverse range of network operations and bring about new applications. For example, by correlating (i) customer opinions posted on web sites with and (ii) response times to a server from various locations, network operators can more quickly identify emerging performance problems and then pinpoint their precise locations. As such, the operators can begin to troubleshoot these problems at an early stage, even before the problems are formally reported by customers. The operators can also discover solutions to these problems by analyzing and classifying the massive number of posts in the operators' communities (e.g., discussions in NANOG list [122]). In addition to the use in fault management, the analysis of big data allows operators to learn customer sentiment in real time and thus to design services and policies that best suit the needs of customers.

4.3. Monitoring data-center networks (cloud)

Data-center networks are often oversubscribed, in order to maximize the utilization of network resources and thus to boost their profits. When a data center is oversubscribed, it can be overloaded. Therefore, a key monitoring objective is to quickly identify overload and performance degradation [116]. If overload is found, then the design operations reallocate overloaded applications to unused resources. The localization of problems can leverage existing methods in fault management (Section 3.5.5), most of which traverse dependency graphs and trace back to likely causes of problems. The metrics to monitor (e.g., latencies through data centers and CPU utilization [117]) and technologies to use (e.g., SNMP and ICMP [118]) can also take advantage of existing solutions.

Nevertheless, the accuracy of monitoring can be improved by carefully considering the characteristics of virtualized infrastructure. Multiple applications can be co-located on the same physical host, and monitoring needs to consider the interactions among these applications and also the interactions between applications and hosts. For example, overload can occur due to contention and interference between the applications [119]. Another characteristic of virtualized infrastructure is that different applications do not share information for security reasons, but sharing of certain monitoring information can benefit performance. If the applications share information about poor performance of a shared storage, the scheduler can postpone access to this storage and does not aggravate the problem [120]. In summary, to better monitor data centers, we need to understand the differences between virtualized infrastructure and traditional, dedicated infrastructure and then to utilize these differences.

4.4. Summary

This section presents issues that are related to multiple layers, among the five layers presented in Section 3. The issues can be resolved by considering the multiple layers together, rather than investigating each layer independently. Section 4.1 explains three major determinants of monitoring quality: (i) reduction of measurement overhead, (ii) accurate construction of network models, and (iii) flexibility in describing measurement objectives. Although these properties often pose trade-offs, intelligent methods exist that improve the three properties altogether. For example, we can allow the description of flexible policies, such that only a selected subset of nodes intensively performs measurements near problematic sites. The result is more efficient and accurate measurement of the problem. Section 4.2 mentions that the average network bandwidth and the number of nodes have been rapidly increasing, and so as the amount of data collected. Therefore, we need methods to efficiently analyze, store, and present the large amount of collected data, such as the aggregation of redundant data and parallel processing. Section 4.3 describes monitoring issues related to the emerging data-center networks. In particular, the overutilization of network resources needs to be quickly identified and mitigated. One solution to this problem is to monitor the contention and interference among virtualized instances.

5. Conclusion

We present open problems in network monitoring and suggest guidelines for future network-monitoring systems. In particular, we identify five major problem-groups: (i) *collection* of data with minimal overhead to collection devices, (ii) *representation* of collected data in consistent formats, (iii) efficient *reporting* of collected data to a management station, (iv) providing higher applications with a diverse range of *analysis* results, and (v) *presentation* of analyzed data to network operators in intuitive ways. We also observe that monitoring operations closely interact with design and deployment operations, and the automation of this interaction can significantly improve the accuracy and efficiency of network monitoring.

In addition to network monitoring, the design operations require more attention from the research community. Accurate design of configuration changes can lead to much fewer problems that surface during monitoring and thus can reduce the time needed to troubleshoot problems. One way to improve the design accuracy is to develop a system that evaluates the correctness of design changes before these changes are deployed in a network. Such an evaluation system needs to be flexible so that the accuracy of various configuration-components can be verified with minimal intervention from network operators (e.g., firewall, routing, VPN, VLAN, and ACL configurations). In other words, adding a new configuration component to the evaluation system should not take operators an excessive amount of time customizing the system. We believe that a combination of design-evaluation systems and improved monitoring can more efficiently reduce network downtime and operational costs.

Acknowledgment

This work was supported by a special research grant from Seoul Women's University (2014).

References

[1] Evaluating high availability mechanisms, Agilent Technologies WhitePaper, 2005. <http://cp.literature.agilent.com/litweb/pdf/5989-4388EN.pdf> (accessed 12.08.13).

[2] Network Lifecycle Management: A Solution Approach to Managing Networks, Hewlett-Packard, White Paper, October 2007.

[3] A.A. Team, Summary of the Amazon EC2 and Amazon RDS Service Disruption in the US East Region, April 2011. <http://aws.amazon.com/ko/message/65648/> (accessed 12.08.13).

[4] M.A. Brown, T. Underwood, E. Zmijewski, The day the Youtube dies, NANOG 43 (2008). June.

[5] Z. Kerravala, As the Value of Enterprise Networks Escalates, so does the Need for Configuration Management, Enterprise Computing and Networking, Yankee Group, 2004.

[6] A. Kind, X. Dimitropoulos, S. Denazis, B. Claise, Advanced network monitoring brings life to the awareness plane, IEEE Commun. Mag. 46 (10) (2008) 140–146.

[7] M. Steinder, A.S. Sethi, A survey of fault localization techniques in computer networks, Elsevier Sci. Comput. Programming 53 (2) (2004) 165–194.

[8] A. Pras, J. Schonwalder, M. Burgess, O. Festor, G.M. Perez, R. Stadler, B. Stiller, Key research challenges in network management, IEEE Commun. Mag. 45 (10) (2007).

[9] J. Schonwalder, A. Pras, J.P. Martin-Flatin, On the future of internet management technologies, IEEE Commun. Mag. 41 (10) (2003) 90–97.

[10] M.J. Wright, Using policies for effective network management, Int. J. Network Manage 9 (2) (1999) 118–125.

[11] R. Mahajan, D. wetherall, T. Anderson, Understanding BGP misconfigurations, in: Proc. ACM SIGCOMM, August 2002, pp. 3–16.

[12] A. Feldmann, J. Rexford, IP network configuration for intradomain traffic-engineering, IEEE Network Mag. 15 (5) (2001) 46–57.

[13] N. Feamster, H. Balakrishnan, Detecting BGP configuration faults with static analysis, in: Proc. USENIX NSDI, May 2005, pp. 43–56.

[14] S. Lee, T. Wong, H.S. Kim, NetPiler: detection of ineffective router configurations, IEEE J. Sel. Areas Commun. Special Issue Network Inf. Conf. 27 (3) (2009) 291–301.

[15] S. Lee, T. Wong, H.S. Kim, Improving manageability through reorganization of routing-policy configurations, Elsevier Comput. Networks 56 (14) (2012) 3192–3205.

[16] S. Lee, H.S. Kim, Correlation, visualization, and usability analysis of routing policy configurations, IEEE Trans. Netw. Serv. Manage. 7 (1) (2010) 28–41.

[17] K.G. Anagnostakis, M. Greenwald, R.S. Ryger, cing: measuring network-internal delays using only existing infrastructure, in: Proc. IEEE INFOCOM, March 2003, pp. 2112–2121.

[18] Clink, <http://allendowney.com/research/clink/> (accessed 12.08.13).

[19] N. Hu, L. Li, Z.M. Mao, P. Steenkiste, J. Wang, Locating internet bottlenecks: algorithms, measurements, and implications, in: Proc. ACM SIGCOMM, October 2004, pp. 41–54.

[20] R.S. Prasad, M. Murray, C. Dovrolis, k. claffy, Bandwidth estimation: metrics, measurement techniques, and tools, IEEE Network Mag. 17 (6) (2003) 27–35. November/December 2003.

[21] S. Saroiu, P.K. Gummadi, S.D. Gribble, Sprobe: a fast technique for measuring bottleneck bandwidth in uncooperative environments, in: Proc. IEEE INFOCOM, June 2002.

[22] B. Floering, B. Brothers, Z. Kalbarczyk, R.K. Iyer, An adaptive architecture for monitoring and failure analysis of high-speed networks, in: Proc. IEEE/IFIP DSN, 2002.

[23] G.L. dos Santos, V.T. Guimaraes, J.G. Silveira, A.T. Vieira, J.A. de Oliveira Neto, R.I.T. da Filho, R. Balbinot, UAMA: a unified architecture for active measurements in IP networks; End-to-end objective quality indicators, in: Proc. IEEE/IFIP IM, 2007.

[24] M. Karsten, J. Schmitt, Packet marking for integrated load control, in: Proc. IEEE/IFIP IM, 2005.

[25] R. Kompella, K. Levchenko, A.C. Snoeren, G. Varghese, Every microsecond counts: tracking fine-grain latencies with a lossy difference aggregator, in: Proc. ACM SIGCOMM, August 2009.

[26] M.J. Luckie, A.J. McGregor, H. Braun, Towards improving packet probing techniques, in: Proc. ACM SIGCOMM IMW, 2001.

[27] S. Machiraju, D. Veitch, A measurement-friendly network (MFN) architecture, in: Proc. ACM SIGCOMM Workshop on INM, 2006.

[28] P. Papageorge, J. McCann, M. Hicks, Passive aggressive measurement with MGRP, ACM SIGCOMM Comput. Commun. Rev. 39 (4) (2009) 279–290.

[29] C. Estan, K. Keys, D. Moore, G. Varghese, Building a better NetFlow, in: Proc. ACM SIGCOMM, 2004.

[30] E.A. Hernandez, M.C. Chidester, A.D. George, "Adaptive sampling for network management", J. Netw. Syst. Manage. 9 (4) (2001).

[31] C. Barakat, G. Iannaccone, C. Diot, Ranking flows from sampled traffic, in Proc. ACM CoNEXT, 2005.

[32] RRDTool. <http://oss.oetiker.ch/rrdtool/> (accessed 12.08.13).

[33] N. Duffield, M. Grossglauser, Trajectory sampling with unreliable reporting, IEEE/ACM Trans. Networking 16 (1) (2008) 37–50.

[34] N.G. Duffield, M. Grossglauser, Trajectory sampling for direct traffic observation, IEEE/ACM Trans. Networking 9 (3) (2001) 280–292.

[35] V. Sekar, M.K. Reiter, W. Willinger, H. Zhang, R.R. Kompella, D.G. Andersen, CSAMP: a system for network-wide flow monitoring, in: Proc. USENIX NSDI, 2008.

[36] A Simple Network Management Protocol (SNMP), RFC-1157, May 1990.

[37] Network Configuration Protocol (NETCONF), RFC-6241, June 2011.

[38] YANG – A Data Modeling Language for the Network Configuration Protocol (NETCONF), RFC-6020, October 2010.

[39] CIM Standards, <http://www.dmtf.org/standards/cim/> (accessed 12.08.13).

[40] IP Flow Information Export (ipfix), <http://datatracker.ietf.org/wg/ipfix/charter/> (accessed 12.08.13).

[41] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, S.C. Diot, Packet-level traffic measurements from the Sprint IP backbone, IEEE Network Mag. 17 (6) (2003) 6–16.

[42] Y. Lin, M.C. Chan, A scalable monitoring approach based on aggregation and refinement, IEEE J. Sel. Areas Commun. 20 (4) (2002) 677–690.

[43] M. Cheikhrouhou, J. Labetoulle, An efficient polling layer for SNMP, in: Proc. IEEE/IFIP NOMS, 2000.

[44] D. Plonka, FlowScan: a network traffic flow reporting and visualization tool, in: Proc. USENIX LISA, 2000.

[45] B. Trammell, C. Gates, NAF: the NetSA aggregated flow tool suite, in: Proc. USENIX LISA, 2006.

[46] S. Romig, The OSU flow-tools package and CISCO NetFlow logs, in: Proc. USENIX LISA, 2000.

[47] J. Sellens, Thresh – a data-directed SNMP threshold poller, in: Proc. USENIX LISA, 2000.

[48] S.S. Kim, A.L. Reddy, NetViewer: a network traffic visualization and analysis tool, in: Proc. USENIX LISA, 2005.

[49] R. Beverly, RTG: A Scalable SNMP Statistics Architecture for Service Providers, in: Proc. USENIX LISA, 2002.

[50] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, F. True, Deriving traffic demands for operational IP networks: methodology and experience, IEEE/ACM Trans. Networking 9 (3) (2001) 265–279.

[51] A. Soule, A. Nucci, R.L. Cruz, E. Leonardi, N. Taft, Estimating dynamic traffic matrices by using viable routing changes, IEEE/ACM Trans. Networking 15 (3) (2007) 485–498.

[52] Y. Zhang, M. Roughan, W. Willinger, L. Qiu, Spatio-temporal compressive sensing and internet traffic matrices, ACM SIGCOMM CCR 39 (4) (2009) 267–278.

[53] M. Crovella, E. Kolaczyk, Graph wavelets for spatial traffic analysis, in: Proc. IEEE INFOCOM, 2003.

[54] K. Papagiannaki, N. Taft, Z.L. Zhang, C. Diot, Long-term forecasting of Internet backbone traffic: observations and initial models, in: Proc. IEEE INFOCOM, 2003.

[55] Y. Zhang, Z. Ge, Finding critical traffic matrices, in: Proc. IEEE/IFIP Dependable Systems and Networks, 2005.

[56] H. Kim, K. Claffy, M. Fomenkov, D. Barman, M. Faloutsos, K. Lee, Internet traffic classification demystified: myths, caveats, and the best practices, in: Proc. ACM CONEXT, 2008.

[57] A. Moore, K Papagiannaki, Toward the Accurate Identification of Network Applications, in: Proc. PAM, March 2001.

[58] T.S. Choi, C.H. Kim, S. Yoon, J.S. Park, B.J. Lee, H.H. Kim, H.S. Chung, T.S. Jeong, Content-aware Internet application traffic measurement and analysis, in: Proc. IEEE/IFIP NOMS, 2004.

[59] S. Sen, O. Spatscheck, D. Wang, Accurate, scalable in-network identification of p2p traffic using application signatures, in: Proc. WWW, 2004.

[60] F. Risso, M. Baldi, O. Morandi, A. Baldini, P. Monclus, Lightweight, payload-based traffic classification: an experimental evaluation, in: Proc. IEEE ICC, 2008.

[61] L. Bernaille, R. Teixeira, Early recognition of encrypted application, in: Proc. PAM, 2007.

[62] T. Karagiannis, K. Papagiannaki, M. Faloutsos, BLINC: multilevel traffic classification in the dark, in: Proc. ACM SIGCOMM, 2005.

[63] K. Xu, Z. Zhang, S. Bhattacharyya, Profiling internet backbone traffic: behavior models and applications, in: Proc. ACM SIGCOMM, 2005.

[64] M. Iliofotou, P. Pappu, M. Faloutsos, M. Mitzenmacher, S. Singh, G. Varghese, Network monitoring using traffic dispersion graphs (tdgs), in: Proc. IMC, 2007.

[65] T. Nguyen, G. Armitage, A survey of techniques for Internet traffic classification using machine learning, IEEE Commun. Surv. Tutorials 10 (4) (2008) 56–76.

[66] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, Y. Guan, Network traffic classification using correlation information, IEEE Trans. Parallel Distrib. Syst. 24 (1) (2013).

[67] Z. Li, R. Yuan, X. Guan, Accurate classification of the Internet traffic based on the SVM Method, in: Proc. IEEE ICC, 2007.

[68] A.W. Moore, D. Zuev, Internet traffic classification using Bayesian analysis techniques, in: Proc. ACM SIGMETRICS, 2005.

[69] C. Estan, S. Savage, G. Varghese, Automatically inferring patterns of resource consumption in network traffic, in: Proc. ACM SIGCOMM, 2003.

[70] M. Baldi, E. Baralis, F. Risso, Data mining techniques for effective and scalable traffic analysis, in: Proc. IEEE/IFIP IM, 2005.

[71] S. Kandula, R. Chandra, D. Katabi, What's going on? Learning communication rules in edge networks, in: Proc. ACM SIGCOMM, 2008.

[72] R.R. Kompella, J. Yates, A. Greenberg, A.C. Snoeren, Detection and localization of network black holes, in: Proc. IEEE INFOCOM, 2007.

[73] I. Cunha, R. Teixeira, N. Feamster, C. Diot, Measurement methods for fast and accurate blackhole identification with binary tomography, in: Proc. ACM IMC, 2009.

[74] P. Bahl, R. Chandra, A. Greenberg, S. Kandula, D. Maltz, M. Zhang, Towards highly reliable enterprise network services via inference of multi-level dependencies, ACM. SIGCOMM CCR 37 (4) (2007) 13–24.

[75] S. Kandula, R. Mahajan, P. Verkaik, S. Agarwal, J. Padhye, P. Bahl, Detailed diagnosis in enterprise networks, ACM SIGCOMM CCR 39 (4) (2009) 243–254.

[76] A. Turner, H.S. Kim, T. Wong, Automatic discovery of relationships across multiple network layers, in: Proc. ACM SIGCOMM Workshop on INM, 2007.

[77] A. Mahimkar, J. Yates, Y. Zhang, A. Shaikh, J. Wang, Z. Ge, C.T. Ee, Troubleshooting chronic conditions in large IP networks, in: Proc. ACM CONEXT, 2008.

[78] R.R. Kompella, J. Yates, A. Greenberg, A.C. Snoeren, IP fault localization via risk modeling, in: Proc. USENIX NSDI, 2005.

[79] S. Kandula, D. Katabi, J. Vasseur, Shrink: a tool for failure diagnosis in IP networks, in: Proc. ACM SIGCOMM MineNet, 2005.

[80] X. Wu, D. Turner, C. Chen, D. A. Maltz, X. Yang, L. Yuan, M. Zhang, NetPilot: automating datacenter network failure mitigation, in: Proc. ACM SIGCOMM, August 2012.

[81] C.J. Tengi, J.M. Roberts, J.R. Crouthamel, C.M. Miller, C.M. Sanchez, autoMAC: a tool for automating network moves, adds, and changes, in: Proc. USENIX LISA, November 2004.

[82] J.R. Crouthamel, J.M. Roberts, C.M. Sanchez, C.J. Tengi, PatchMaker: a physical network patch manager tool, in: Proc. USENIX LISA, November 2004.

[83] R.M. Oliveira, S. Lee, H.S. Kim, Automatic detection of firewall misconfigurations using firewall and network routing policies, in: IEEE DSN Workshop on Proactive Failure Avoidance, Recovery, and Maintenance (PFARM), Lisbon, Portugal, June 2009.

[84] D. Caldwell, A. Gilbert, J. Gottlieb, A. Greenberg, G. Hjalmtysson, J. Rexford, The cutting EDGE of IP router configuration, in: Proc. Workshop on Hot Topics in Networks (HotNets), November 2003.

[85] K. Levanti, H.S. Kim, T. Wong, NetPolis: Modeling of inter-domain routing policies, in: Proc. IEEE GLOBECOM, 2008.

[86] M. Soni, V.R. Madduri, M. Gupta, P. De, Tracking configuration changes proactively in large IT environments, in: Proc. IEEE/IFIP NOMS, April 2012.

[87] T. Oetiker, MRTG: the Multi Router Traffic Grapher, in: Proc. USENIX LISA, 1998.

[88] SNAPP, <http://sourceforge.net/projects/snapp/> (accessed 12.08.13).

[89] J. Oberheide, M. Goff, M. Karir, Flamingo: visualizing internet traffic, in: Proc. IEEE/IFIP NOMS, 2006.

[90] W. Yurcik, Visualizing NetFlows for security at line speed: the SIFT tool suite, in: Proc. USENIX LISA, 2005.

[91] F. Fischer, F. Mansmann, D.A. Keim, S. Pietzko, M. Waldvogel, Large-scale network monitoring for visual analysis of attacks, in: Proc. IEEE Symposium on Visualization for Cyber Security (VizSec), 2008.

[92] A. Yelizarov, D. Gamayunov, Visualization of complex attacks and state of attacked network, in: Proc. IEEE Symposium on Visualization for Cyber Security (VizSec), 2009.

[93] B. Choi, S. Moon, R. Cruz, Z. Zhang, C. Diot, Practical delay monitoring for ISPs, in: Proc. ACM CONEXT, 2005.

[94] A.G. Prieto, R. Stadler, A-GAP: an adaptive protocol for continuous network monitoring with accuracy objectives, IEEE Trans. Netw. Serv. Manage. 4 (1) (2007) 2–12.

[95] M. Dilman, D. Raz, Efficient reactive monitoring, IEEE J. Sel. Areas Commun. 20 (4) (2002) 668–676.

[96] J. Jiao, S. Naqvi, D. Raz, B. Sugla, Toward efficient monitoring, IEEE J. Sel. Areas Commun. 18 (5) (2000) 723–732.

[97] D. Breitgand, D. Dolev, D. Raz, G. Shaviner, Facilitating efficient and reliable monitoring through HAMSA, in: Proc. IEEE/IFIP IM, 2003.

[98] Y. Bejerano, R. Rastogi, Robust monitoring of link delays and faults in IP networks, IEEE/ACM Trans. Networking 14 (5) (2006) 1092–1103.

[99] C. Chaudet, E. Fleury, I.G. Lassous, H. Rivano, M. Voge, Optimal positioning of active and passive monitoring devices, in: Proc. ACM CoNEXT, 2005.

[100] L. Li, M. Thottan, B. Yao, S. Paul, Distributed network monitoring with bounded link utilization in IP networks, in: Proc. IEEE INFOCOM, 2003.

[101] G. Huang, C. Chang, C. Chuah, B. Lin, Measurement-aware monitor placement and routing: a joint optimization approach for network-wide measurements, IEEE Trans. Netw. Serv. Manage. 9 (1) (2012) 48–59.

[102] Y. Choi, I. Hwang, In-service QoS monitoring of real-time applications using SM MIB, Int. J. Network Manage. 15 (1) (2005) 31–42.

[103] G.A. Winters, D.A. Muntz, T.J. Teorey, Using RMON matrix group extensions to analyze internetworking problems, J. Netw. Syst. Manage. 6 (2) (1998) 179–196.

[104] L. Yuan, C.-N. Chuah, P. Mohapatra, ProgME: towards programmable network measurement, in: Proc. ACM SIGCOMM, 2007.

[105] P. de Meer, A. La Corte, A. Puliafito, O. Tomarchio, Programmable agents for flexible QoS management in IP networks, IEEE J. Sel. Areas Commun. 18 (2) (2000) 256–267.

[106] A. Liotta, G. Pavlou, G. Knight, Exploiting agent mobility for large-scale network monitoring, IEEE Network Mag. 16 (3) (2002) 3–15.

[107] N. Spring, D. Wetherall, T. Anderson, Scriptroute: a public internet measurement facility, in: Proc. USITS, 2002.

[108] E.P. Duarte, M.A. Musicante, H.H. Fernandes, ANEMONA: a programming language for network monitoring applications, Int. J. Network Manage. 18 (4) (2008).

[109] J. Francois, C. Wagner, R. State, T. Engel, SAFEM: scalable analysis of flows with entropic measures and SVM, in: Proc. IEEE/IFIP NOMS, April 2012.

[110] N. Bonelli, A.D. Pietro, S. Giordano, G. Procissi, On multi-gigabit packet capturing with multi-core commodity hardware, in: Proc. PAM, 2012.

[111] T. Samak, D. Gunter, V. Hendrix, Scalable analysis of network measurements with Hadoop and Pig, in: Proc. IEEE/IFIP NOMS, April 2012.

[112] J. Dean, S. Ghemawat, Mapreduce: simplified data processing on large clusters, Commun. ACM 51 (1) (2008) 107–113.

[113] D. Simoncelli, M. Dusi, F. Gringoli, S. Niccolini, "Stream-monitoring with BlockMon: convergence of network measurements and data analytics platforms", ACM SIGCOMM CCR 43 (2) (2013) 30–35. April 2013.

[114] L. Neumeyer, B. Robbins, A. Nair, A. Kesari, S4: distributed stream computing platform, in: Proc. IEEE International Conference on Data Mining Workshops (ICDMW), 2010.

[115] Storm, <http://storm-project.net> (accessed 12.08.13).

[116] S.A. Baset, L. Wang, C. Tang, Towards an understanding of oversubscription in cloud, in: Proceedings of USENIX Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE), 2012.
[117] P. Singh, M. Lee, S. Kumar, R.R. Kompella, Enabling flow-level latency measurements across routers in data centers, in: Proceedings of USENIX Hot-ICE, 2011.
[118] S. Sarkar, R. Mahindru, R.A. Hosn, N. Vogl, H.V. Ramasamy, Automated incident management for a platform-as-a-service cloud, in: Proceedings of USENIX Hot-ICE, 2011.
[119] H. Kang, X. Zhu, J.L. Wong, DAPA: diagnosing application performance anomalies for virtualized infrastructure, in: Proceedings of USENIX Hot-ICE, 2012.
[120] F. Dinu, T.S. Eugene Ng, Synergy2Cloud: introducing cross-sharing of application experiences into the cloud management cycle, in: Proceedings of USENIX Hot-ICE, 2012.
[121] B. Trammell, E. Boschi, L. Mark, T. Zseby, A. Wagner, Specification of the IP Flow Information Export (IPFIX) File Format, RFC-5655, October. 2009.
[122] NANOG (North American Network Operators' Group) Mail List, <http://www.nanog.org/list> (accessed 16.12.13).

Sihyung Lee received the B.S. (with *summa cum laude*) and M.S. degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST) in 2000 and 2004, respectively, and a Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University (CMU) in 2010. He then worked at IBM TJ Watson Research Center as a post-doctoral researcher. He is currently an assistant professor at Seoul Women's University in the Department of Information Security. His research interests include the management of large-scale network configurations and sentiment-pattern mining from social network traffic. In particular, he has been working with production-network configurations and measurement data for over 7 years.

Kyriaki Levanti received the Diploma degree in Electrical and Computer Engineering from National Technical University of Athens (NTUA), Greece, in 2005, and a Ph.D. degree in Electrical and Computer Engineering from Carnegie Mellon University (CMU), Pittsburgh, PA, in 2012. She worked in the Distributed Systems Group at IBM T.J. Watson Research Center, Hawthorne, NY, in the summer of 2008. She is currently working at Amazon, Seattle, as a software engineer. Her research interests include Internet routing, network management and measurement.

Hyong S. Kim received the B.Eng. (*Hons*) degree in Electrical Engineering from McGill University in 1984, and the M.A.Sc. and Ph.D. degrees in Electrical Engineering from University of Toronto, Canada, in 1987 and 1990, respectively. He has been with Carnegie Mellon University (CMU), Pittsburgh, PA, since1990, where he is currently the Drew D. Perkins Chaired Professor of Electrical and Computer Engineering. His primary research areas are advanced switching architectures, fault-tolerant, reliable, and secure network architectures, and network management and control.