

Abdun Naser Mahmood, Christopher Leckie,
Jiankun Hu, Zahir Tari,
and Mohammed Atiquzzaman

Contents

20.1 Fundamentals of Network Traffic Monitoring and Analysis	384
20.1.1 What Are the Traffic Measurement Problems?	384
20.1.2 Traffic Matrix Measurement	385
20.1.3 Traffic Volume Measurement	385
20.1.4 Traffic Dynamics Measurement	385
20.1.5 Traffic Mixture Measurement	386
20.2 Methods for Collecting Traffic Measurements	386
20.2.1 Passive vs. Active Measurements	386
20.2.2 Network Data Acquisition Cards	388
20.2.3 Packet Traces	388
20.2.4 NetFlow Records	389
20.3 Analyzing Traffic Mixtures	390
20.3.1 Monitoring Pre-Defined, Coarse Aggregates of Traffic Volume ..	390
20.3.2 Monitoring Significant Aggregates of Traffic Volume	392
20.3.3 Monitoring Significant Changes in Traffic Volume	393
20.3.4 Frequency-Based Clustering Using Frequent Itemsets	394
20.4 Case Study: AutoFocus	395
20.5 How Can We Apply Network Traffic Monitoring Techniques for SCADA System Security?	399
20.5.1 SCADA Systems	399
20.5.2 SCADA Security Issues	399
20.5.3 Protecting SCADA Systems by Using Network Traffic Monitoring	400
20.6 Conclusion	401
References	402
The Authors	404

The problem of monitoring and characterizing network traffic arises in the context of a variety of network management functions. For example, consider the five functions defined in the OSI Network Management Framework [20.1], i.e., configuration management, performance management, fault management, accounting management and security management. Traffic monitoring is used in configuration management for tasks such as estimating the traffic demands between different points in the network, so that network capacity can be allocated to these demands. In performance management, traffic monitoring can be used to determine whether the measured traffic levels exceed the allocated network capacity, thus causing congestion or delays. When a fault occurs in the network, traffic monitoring is used in fault management to help locate the source of the fault, based on changes in the traffic levels through the surrounding network elements. In accounting management, traffic monitoring is needed to measure the network usage by each customer, so that costs can be charged accordingly in terms of the volume and type of traffic generated. Finally, network traffic monitoring can be used in security management to identify unusual traffic flows, which may be caused by a denial-of-service attack or other forms of misuse.

SCADA systems are widely used for monitoring and controlling industrial systems including power plants, water and sewage systems, traffic control, and manufacturing industries. The security of SCADA networks is an important topic today due to the vital role that SCADA systems play in our national lives in providing essential utility services. Pervasive Internet accessibility at industrial workplaces increases the vulnerabilities of SCADA systems because this

makes it possible for a remote attacker to gain control of, or cause disruption to the critical functions of the network.

In this chapter, fundamentals of network traffic monitoring management have been introduced in a systematic framework. Advanced technologies have been studied based on published literature including our own published research work [20.2]. Application of network traffic monitoring management to SCADA system security has been investigated. This chapter intends to be a comprehensive reference in the field of network traffic monitoring management. It can be used as a reference for academic researchers and also as a suitable textbook reference for senior undergraduate students and post-graduates for networking management and network security courses.

This chapter spans the fields of network traffic analysis and data mining, which are both extensive fields in their own right. In order to provide a focus, we first describe the relevant background to our problem in network traffic analysis, and then describe work from the data mining community that is related to this problem.

We begin in Sect. 20.1 by describing the general types of traffic analysis problems that arise in the context of managing the Internet. In particular, we emphasize the problems of measuring traffic volumes and traffic mixtures. Traffic volume measurements can help identify large flows that are important because of their impact on provisioning, accounting and performance management of the network. On the other hand, traffic mixture analysis helps in understanding the complex nature of the traffic and identifies patterns in usage that may be useful for fault detection and security management.

In Sect. 20.2, we then describe the relevant methods for collecting the raw observations for network traffic data. In the context of network traffic data, we limit our survey of the types of traffic data to packet headers (Sect. 20.2.3) and NetFlow traces (Sect. 20.2.4), and exclude any survey of packet payload analysis for identifying patterns of user behavior. In payload analysis the content of the packets are analyzed to reveal low-level information about the nature of the traffic. However, there are two problems associated with trying to read packet contents for analysis. First, due to an increase in privacy and security concerns many protocols now support cryptographic measures to prevent man-

in-the-middle attacks and unwanted interception of data over insecure media, thus making the packet payload unavailable for analysis. Second, even if the packet payload is available as plain text or decrypted for analysis, processing the payload is resource intensive and not scalable with the rate that packets arrive for medium to fast connections. Consequently, we limit our attention to packet headers and NetFlow traces.

In Sect. 20.3, we focus on related work to the problem of analyzing the mixture of traffic on a network, which is the focus of our chapter. Of particular relevance is the problem of monitoring significant aggregates of traffic, in order to identify the types of aggregate flows that are utilizing the network. One particular approach that is widely used in this context is frequent itemset mining. In Sect. 20.4, we discuss frequent itemset clustering for traffic mixture analysis. In particular, we examine how a frequent itemset clustering tool, AutoFocus [20.3], generates traffic clusters based on uni-dimensional and multi-dimensional frequent itemset clustering. We also analyze its space and time complexity both theoretically and with the help of an illustration. Finally, in Sect. 20.5 we describe the architecture of a SCADA network and identify key sensor positions for monitoring network traffic to and from the SCADA network.

20.1 Fundamentals of Network Traffic Monitoring and Analysis

20.1.1 What Are the Traffic Measurement Problems?

Traffic measurement is a well-established field of telecommunications research. Early work in this field (e.g. [20.4–6]) focused on the circuit-switched telephone network. In this environment, information about the duration of a call, its origin and destination points, and its route are usually well-defined, and the centrally managed switching and signaling infrastructure provide a platform for collecting this network traffic data.

In contrast, the Internet is a packet-based and highly decentralized network. The design of the Internet has aimed to minimize the amount of higher layer information and connection state data that needs to be kept within the network layer. When

coupled with the highly decentralized structure of the Internet, this has created major challenges for network managers of IP networks. If users experience packet delay or loss, there is no intrinsic support to identify the route those packets took. This creates a challenge for effective performance and fault management. Similarly, it can be difficult to analyze patterns of customer usage because service information is kept in application clients or servers, rather than in the network.

As a consequence, in many network management functions we are forced to infer patterns of user activity indirectly, by analyzing the type of data that is directly available to network operators – namely network traffic traces. This need to infer patterns of user activity has stimulated research into a range of new traffic analysis problems. We divide these traffic analysis problems into four main categories: traffic matrix, traffic volume, traffic dynamics and traffic mixture measurement. Let us summarize the general problem in each case, and highlight our focus on traffic mixture measurement.

20.1.2 Traffic Matrix Measurement

The aim of traffic matrix measurement is to estimate the volume of traffic between origin and destination points in the network. It is used for capacity planning, provisioning network resources and for assessing the effect of network faults on network capacity. General approaches to this problem include network tomography and direct measurement. Network tomography [20.7] aims to indirectly infer end-to-end traffic demands based on traffic measurements on each link in the network, for example, using Simple Network Management Protocol (SNMP) link byte counts [20.8]. This is an under-constrained problem, and numerous approaches have been proposed [20.9–11] to provide additional prior information about where traffic is likely to be headed. In contrast, direct measurement maintains a digest of traffic flows at each origin point [20.12]. These digests are then merged at a central point to find the end point of each flow. The challenge here is to find a method of compressing digests that minimizes the memory requirements at origin-destination points, without significantly reducing accuracy. Configuration management relates to the monitoring of the state of resources and the relationships among resources.

20.1.3 Traffic Volume Measurement

The aim of traffic volume measurement is to determine the total traffic sent or received in a network. Of particular interest is the problem of measuring network usage by consumers. This involves aggregating the total byte or packet count for each source IP address. This type of measurement has become important for accounting management as Internet Service Providers (ISP) have moved from time-based accounting to usage-based accounting of customer charges [20.13]. Traffic volume measurement is also used in performance management and security management to identify heavy users of the network, who may be causing congestion in the network. For example, Roh and Yoo [20.14] propose measuring the ratio of packet count to byte count as a measure to identify abnormal flows. There are several existing tools for traffic volume analysis [20.15]. Some tools [20.3, 16] show the changes in traffic with graphs, e.g., flow-scan [20.17]. Other tools provide “top K reports” of heaviest usage, such as cflowd [20.18] and flow-tools [20.17]. These tools provide visual clues of changes in user behavior at a very high level, for example, by providing a graphical report of IP addresses that are sending the most traffic. A problem with this approach to reporting is that it tells us nothing about sources that send only a small volume of traffic. If these small flows are combined, then they may form a large proportion of the overall traffic. Consequently, these trends may be overlooked unless we can identify relevant patterns among traffic flows. Moreover, graphical tools generally cannot cope well with visualizing traffic with high dimensions, and fail to generalize any underlying patterns. Thus, there is a need for monitoring techniques that can aggregate traffic by attributes other than IP address alone.

20.1.4 Traffic Dynamics Measurement

The aim of monitoring traffic dynamics is to measure the temporal variation in Internet traffic. Knowledge of variation in traffic load is important in configuration management in order to adequately dimension networks. For example, robust estimation of traffic variation can be used to determine the size of buffers, or the extent to which links need to be over-dimensioned [20.19]. Since traditional Poisson models for traffic arrivals fail to account for the

burstiness of Internet traffic [20.20], there has been considerable interest in empirical models based on traffic measurements [20.21]. In performance management, monitoring traffic dynamics is used to test the stability of the network [20.22, 23]. The types of traffic metrics of interest include packet delay, packet loss, and the available bandwidth of bottleneck links [20.24]. In contrast with the problem of measuring traffic dynamics, our focus is on the challenge of monitoring the volume and mixture of flows within a given sample of network traffic.

20.1.5 Traffic Mixture Measurement

As mentioned before, when traffic volume data is aggregated over time it can reveal important features of network usage for performance and security management. Bradford et al. [20.25] studied aggregated traffic volume and showed that signal analysis on data aggregation at certain levels of network traffic helps distinguish among four broad classes of network anomalies, namely, outages, flash crowds, attacks and measurement failures. Kim et al. [20.26] suggest a similar technique for traffic anomaly detection based on analyzing correlations of destination IP addresses in outgoing traffic. This address correlation data is modeled using a discrete wavelet transformation to detect anomalies. Estan et al. [20.3] address the problem of finding patterns in network traffic by proposing a frequent itemset mining algorithm. Their tool, called AutoFocus [20.27], describes the traffic mix on a network link by using textual reports as well as time series plots. It also produces concise reports that can show general trends in the data. In Sect. 20.4, we discuss frequent itemset mining of network data in more detail. Cormode et al. [20.28, 29] have argued that building an exact multidimensional lattice is prohibitively expensive and offer approximate count solutions for a data stream environment. Kim et al. [20.30] use the combination of rule-based flow header detection and a traffic aggregation algorithm. Chhabra et al. [20.31] propose a randomized algorithm that is similar to the technique of Estan et al. [20.3], which aggregates flows with similar field values to yield signatures of network traffic.

In this chapter, we focus on this problem of traffic mixture measurement. A major issue with these techniques is that they are computationally intensive, and hence do not scale well when

analyzing large volumes of traffic. Some of the other works which deal with minimizing the effect of large datasets includes the use of sampling [20.32, 33], flow histogram analysis [20.34], and sketches [20.35].

20.2 Methods for Collecting Traffic Measurements

The input to any traffic analysis system is the raw traffic measurements that can be collected from the network of interest. These include low-level traces of individual packets, as well as slightly higher-level traces of flows, which corresponds to a sequence of packets with common origin and destination points. These are usually collected in a passive manner by observing the existing traffic on a network. In some cases, however, it can be preferable to actively inject traffic into the network in order to observe the effect of the network and other traffic on this injected traffic. In this section, we provide a brief summary of the main approaches to collecting network traffic measurements, and highlight the focus of our research. We begin by comparing passive and active measurements in Sect. 20.2.1. We then outline in Sect. 20.2.2 how passive measurements can be made using network data acquisition cards. In Sects. 20.2.3 and 20.2.4, we then give examples of the main types of traces that can be collected using passive measurement, namely network packet traces and network flow traces.

20.2.1 Passive vs. Active Measurements

Passive Measurement: In passive measurement, network packets are logged and analyzed for various network characteristics. A monitor placed on a network link passively observes the network traffic and collects observations in the form of packet statistics and packet traces. Different applications use this information to infer various characteristics of the network, for example, passive measurements are used to calculate various performance metrics [20.36–40], and understand protocol behavior [20.37, 41]. Benko et al. [20.36] study the end-to-end loss of TCP packets through passive traffic monitoring. They estimate loss ratios by analyzing the patterns of the observed TCP sequence

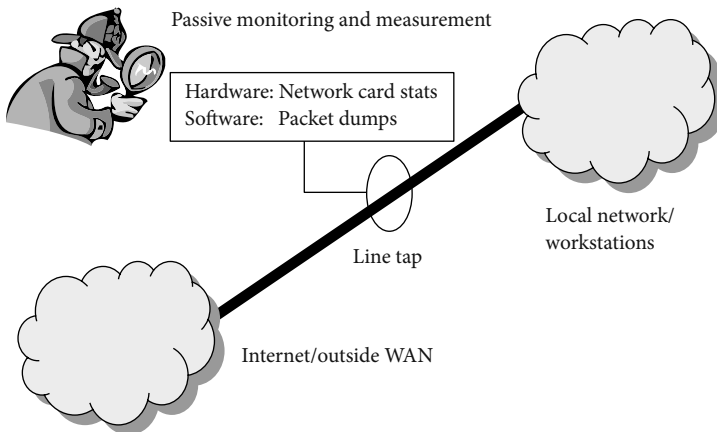


Fig. 20.1 Passive monitoring of network data. The *bottom cloud* represents the outside network and the *top cloud* represents the local network. The monitor sits in the *middle* and collects the data from the network link

numbers. Jaiswal et al. [20.38] estimates the *health* of a TCP connection by passively measuring the number of TCP sequences that are out of order, i.e., non-increasing sequences, and use this to infer the cause behind reordering, loss and duplication of packets. In a different paper, Jaiswal et al. [20.37] also use passive monitoring methodology to infer the *congestion window* and the *connection round trip time* (RTT) of a TCP transmission. Understanding the distribution of RTT is important in buffer provisioning, queue management, and detecting traffic congestion. Jiang et al. [20.39] proposes two techniques to estimate the RTT distribution from unidirectional TCP packets going from the origin to the destination, and the TCP responses from the destination to the origin.

We can broadly summarize the techniques used in passive measurement into two categories based on the amount of data they retain. The first type of passive measurement keeps some statistics about packets and flows, for example, packet count, byte count, and flow count over different periods. The measurements are used in various network management functions [20.42] including optimizing bandwidth utilization, preventing link saturation, and provisioning an increase in bandwidth. This kind of measurement can be used at line speeds because of the low overhead of keeping the statistics. The second type of passive measurement looks into the packets and copies part or all of the packets for later analysis. These trace files are useful for computationally intensive analysis after suitable processing of the data, which may include anonymization of the sensitive information present in the data.

Figure 20.1 illustrates passive monitoring of network traffic between an organization and its outside world. The line tap indicates the passive monitoring of packets using hardware devices and software applications. There can be three levels of data acquisition from the network link: the first is a dedicated network data acquisition card which collects packets or statistics at line speeds; next are router logs in the form of NetFlow records, where flow headers are collected at regular intervals and exported to a workstation for later analysis; and the last is a complete trace in the form of packet dumps.

Active Measurements: In contrast to passive measurement, probe packets can be sent across the network to measure some aspects of dynamic traffic behavior, such as packet delays and loss. Packets are sent from one network access point to another and marked at transition points such as routers in order to measure time delays and the rate of packet loss. For example, the widely known *ping* [20.43] utility sends ICMP *echo* packets for estimating network latency, the *traceroute* [20.44] utility reports routing paths between end points, and *pathchar* [20.45] tool is used for estimating latency and link capacity along a network path. These methods are clearly intrusive, in a sense, because they may also affect the measurement data being collected. Sometimes these utilities are used by malicious users to create DoS attacks. An example of such an attack involving an active measurement tool is the well known *Ping-of-Death* attack, where an attacker overwhelms a target with continuous ping probes until the target is incapacitated [20.46]. Another potential problem with active measurement is the decentralized nature of the In-

ternet. It is required, as a matter of etiquette or sometimes as a matter of law, that concerned network administrators be advised prior to any attempt of actively measuring network traffic data that either terminate at or go through their system. In order to discourage attempts of such “intrusive” measurement some organizations set up rules at their routers to drop or reject unwanted probe packets. In this chapter, we focus on the problem of analyzing network traffic traces that have been collected using passive measurements. Let us now give some examples of how these traces can be collected, as well the context of those traces.

20.2.2 Network Data Acquisition Cards

An increasingly popular method for capturing traffic traces from high speed networks is to use network data acquisition cards. Network cards connect directly to the transmission medium and collect the network traffic at line speeds without distorting the traffic. They have an advantage over using packet capture in routers, because when routers are used to replicate or divert traffic it can overload the internal communications channels within the router. Another example of distortion can occur when an Ethernet switch is used as a repeater and it arbitrarily delays the traffic due to buffering [20.47].

Figure 20.2 shows the network measurement card, called DAG, developed originally by the WAND network research group [20.48] of the University of Waikato and now made by Endace

Technologies [20.49]. The card attaches itself to the physical transmission medium and is able to capture the network traffic at the line speed. At the heart of the device is a large Field Programmable Gate Array (FPGA) that is used to (1) generate accurate timestamps with the help of external GPS and clock devices, (2) transform data from the physical layer into a form that is suitable for the PCI interface, and (3) filter and pre-process incoming data with the help of the processor and RAM. The FPGA allows the card to be reprogrammable for different types of networks. The GPS antenna and the local clock provide accurate time information on collected traffic information. An important advantage of these types of cards is that they can provide a limited functionality to select or filter flows or packets of interest, based on a simple specification of the relevant traffic attributes.

20.2.3 Packet Traces

One form of traffic data that can be captured by data acquisition cards is a packet trace. At a low level, a network device communicates with another by sending and receiving data in packets. Although the information contained in the packets may change for many reasons including protocol and routing strategy, the basic elements include a header section and a payload section. The header section has various fields including source and destination addresses according to the specified protocol, source and destination ports, error and flow control infor-

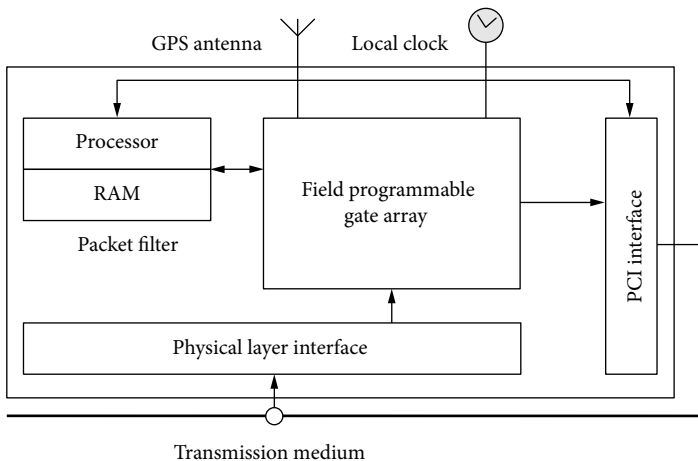


Fig. 20.2 WAND network research group’s DAG network data access card. The processor and RAM are optional and can be used for packet filtering at high line speeds. The FPGA allows the card to be reprogrammable for different types of network. PCI interface connects to the PC. The GPS and local clock allow accurate time stamping on collected traffic information

Table 20.1 An example of a traffic packet trace showing the different fields in the packet header in tabular form. The actual data contained in the packets is not shown

TIME STAMP	PROTOCOL	SRC IP_ADDRESS	SRC PORT	DST PORT	DST IP_ADDRESS	TCP SIZE	TCP SEQ	ACK
0	IP/TCP	127.246.129.64	80	1060	27.86.12.4	40	920,641	412,791
14,966	IP/TCP	161.77.104.57	80	7410	27.86.12.4	508	410,104	32,779
15,015	IP/TCP	91.82.74.90	80	1105	91.82.59.75	40	2,816,846	7726
22,090	IP/TCP	19.27.2.59	80	1140	26.37.13.44	40	1,010,185	14,762
22,126	IP/TCP	82.127.55.91	80	1291	19.74.87.6	40	9,557,082	50,482
29,960	IP/TCP	61.77.104.57	80	3741	27.86.12.4	40	985,526	58,006
29,960	IP/TCP	19.74.87.6	1291	80	82.127.55.91	1500	653,402	57,082
31,724	IP/TCP	19.74.87.6	1291	80	82.127.55.91	1500	654,862	57,082
36,055	IP/TCP	12.84.9.17	80	1125	19.74.87.28	311	857,517	89,873
36,279	IP/TCP	12.84.9.17	80	1126	19.74.87.28	271	857,661	3293
37,181	IP/TCP	207.84.92.183	5190	1207	98.54.73.39	40	64,202	9407
41,731	IP/TCP	99.81.77.33	1116	80	42.6.74.91	40	1,062,629	68,778

mation and time stamping. The payload is the data created by the application, which initiates the communication. Sometimes the data needs to be split into multiple packets because of packet size restrictions imposed by various networks. Table 20.1 is an example of a TCP/IP packet header. Notice the sequence and acknowledgement numbers that allow the router to reconstruct higher level sessions from a series of packets.

While packet traces produce a detailed view of activity on a network, their size can be overwhelming in large networks. An alternative approach is to collect a trace of the flows that generated these packets. Next, we describe the contents of a network flow trace.

20.2.4 NetFlow Records

Today's high speed networks create a challenge for network operators in terms of the storage and processing facilities needed to cope with the high vol-

umes of packet traces that can be generated by these networks. To address these problems, Cisco implemented the NetFlow protocol for collecting IP traffic information from their routers [20.51]. The NetFlow protocol is now an open standard and because of its simplicity has been adopted by other network equipment vendors such as Juniper Networks (who calls it JFlow [20.52]) and Huawei Technology (who calls it NetStream [20.53]). Because of its popularity it has been accepted as an industry standard by the IETF called Internet Protocol Flow Information eXport or IPFIX [20.54].

As can be seen from Table 20.2, a NetFlow record consists primarily of a five tuple: source IP address, destination IP address, protocol, source port and destination Port. A NetFlow record is defined as a unidirectional sequence of packets sharing the same values for these attributes. The router maintains a table of existing flows in memory and creates a new one whenever a new source IP address originates a connection to a destination IP address. It

Table 20.2 An example of a NetFlow trace showing the different fields in the flow trace in tabular form

PROTOCOL	SRC IP_ADDRESS	SRC PORT	DST PORT	DST IP_ADDRESS	FLOW SIZE
IP/TCP	127.246.129.64	80	1060	27.86.12.4	40
IP/TCP	161.77.104.57	80	7410	27.86.12.4	508
IP/TCP	91.82.74.90	80	1105	91.82.59.75	40
IP/TCP	19.27.2.59	80	1140	26.37.13.44	40
IP/TCP	82.127.55.91	80	1291	19.74.87.6	40
IP/TCP	61.877.104.57	80	3741	27.86.12.4	40
IP/TCP	19.74.87.6	1291	80	82.127.55.91	3000
IP/TCP	12.84.9.17	80	1125	19.74.87.28	582
IP/TCP	207.84.92.83	5190	1207	98.54.73.39	40
IP/TCP	99.81.77.33	1116	80	42.6.74.91	40

continues to update the counters for packet numbers and sizes until the last of the packets in the transmission has been received or until it reaches a timeout. NetFlow helps reduce the size of the network data generated by aggregating on several fields including packet and flow size counters. This helps identify some of the larger flows that may be causing bottlenecks in the system or that may be the result of a DoS attack.

In this chapter, we focus on the problem of finding patterns of traffic in a given trace of network flow records. In the next section, we consider the types of data mining problems that arise in the context of analyzing this type of data.

20.3 Analyzing Traffic Mixtures

As discussed in Sect. 20.1.5, our focus is on the problem of analyzing traffic measurements in order to characterize the mixture of different types of aggregate flows on a network. In the literature, a number of different methods have been proposed to address different aspects of this problem. We categorize this previous research in terms of (1) monitoring pre-defined, coarse aggregate of traffic volume, (2) monitoring significant aggregates of traffic volume, and (3) monitoring significant changes in traffic volume.

Let us now summarize the related research in each of these areas, and highlight the relationship to our research.

20.3.1 Monitoring Pre-Defined, Coarse Aggregates of Traffic Volume

An aggregate flow is a set of raw flows that have the same value for a subset of their attributes, e.g., a set of flows between the same source address and destination address, or a set of flows with a same protocol field value. One approach to analyzing the mixture of traffic on a network is to measure the volume of traffic for a set of pre-defined aggregate flows, e.g., the traffic volume from a set of source addresses, using specific protocols of interest. The advantage of this approach is that the set of aggregates that needs to be monitored is static. Hence this general approach has been used in earlier systems where computational resources are limited. We now consider some examples of this approach based on *SNMP* data collection, as well as a number of flow based tools.

SNMP based coarse aggregates: The Simple Network Management Protocol (SNMP) is an application layer protocol for monitoring routers and other network devices. It has an agent/manager model, as shown in Fig. 20.3, where the agent entity uses

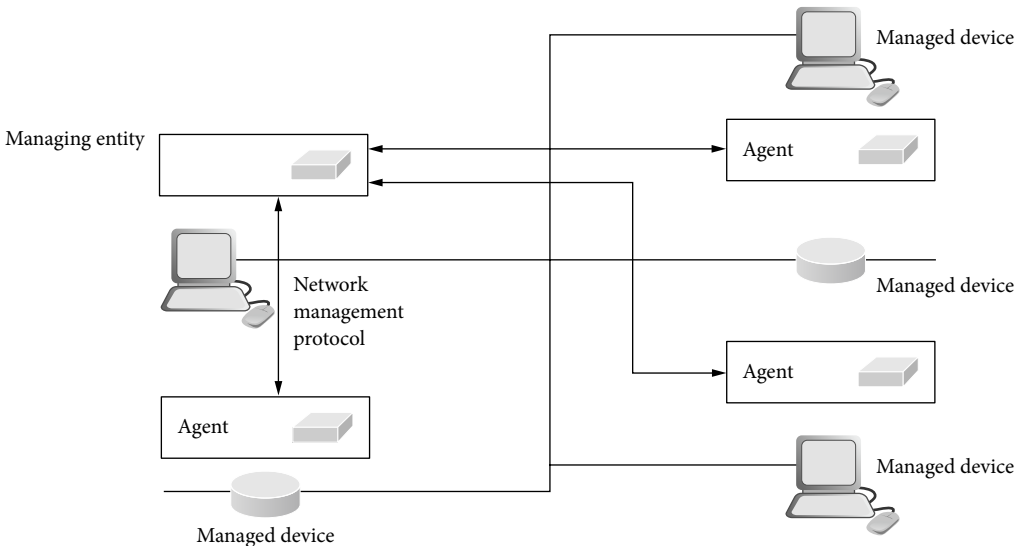


Fig. 20.3 An illustration of the main entities in the Simple Network Management Protocol (SNMP). The managing entity gathers information from the agents about the managed devices using the management protocol (figure is based on [20.50])

a Management Information Base (MIB) to store information about a managed device and a set of commands to exchange information with the managing entity. The MIB has a tree structure where the variables of interest are represented as the leaves of the tree. An object identifier is a numeric tag that distinguishes one variable from the other. A managed entity can accumulate counts for these predefined variables in the MIB, and the values of these variables can be accessed by the managing entity using SNMP.

Since most SNMP based devices have limited storage, they can only give high-level information on network usage, for example, interface bandwidth and link utilization. However, this information is important since administrators can use it to constantly monitor the availability of the link, the link usage and some high-level network usage characteristics. Examples of popular tools that use SNMP data are *MRTG* [20.55] and *Cricket* [20.56]. Next, we mention briefly about the functionality of MRTG as an example.

MRTG [20.55] (or *Multi Router Traffic Grapher*) is a popular traffic visualization tool for SNMP data. It continuously queries each agent to retrieve measurement data and plots them to give a graphical representation of the traffic, as shown in Fig. 20.4. MRTG stores the information in a Round Robin Database (RRD) [20.57], developed by the same author of MRTG, which keeps the database small by an efficient implementation of binary log files as well as on-demand generation of graphs.

NetFlow based measurement: In contrast to the high level statistics provided by SNMP, NetFlow based tools offer finer granularity and greater insight into the traffic data. Some of the popular tools

for collecting and analyzing NetFlow data are *Flow-tools* [20.17], *FlowScan* [20.58], *Fluxoscope* [20.59] and *ntop* [20.60]. In the following we use *Flow-tools*, *FlowScan* and *ntop* as examples of how NetFlow data can be analyzed.

Flow-tools [20.61] is a collection of programs for collecting, transferring, processing, and generating reports from NetFlow data. Figure 20.5 shows an example of a report generated by *Flow-tools*. Report *A* shows the coarse aggregates by the protocol field. Report *B* shows more detailed information about the nature of the traffic flows by including fields such as the total number of flows, average flow size, average packet size and average number of packets per flow.

FlowScan is a NetFlow visualization tool that uses a collection of scripts to produce graphs of network traffic. *FlowScan* also uses the *RRDTool* database to store numerical time-series data as shown in Fig. 20.6. Such a graph often reveals interesting patterns of usage. For example, the campus traffic peaks in the late evening and has a low point around 6:00 AM. The fact that the total outbound traffic is more than the total inbound traffic and the presence of a high proportion of HTTP data (shown in red) in the outbound traffic may indicate that the campus webserver is very busy.

The large purple area indicates the presence of Napster data content and shows that it is comparable to the amount of web and FTP traffic present in the network.

The *ntop* tool is a simple yet powerful tool that reports the top network users by quickly identifying those hosts that are currently using most of the available network resources. The *ntop* tool is open source, and is similar in design to the UNIX *top* tool. The types of information recorded by *ntop* are: statistics

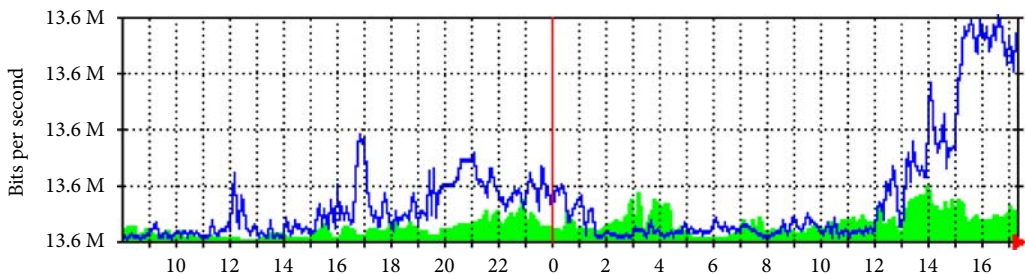


Fig. 20.4 A screen dump of MRTG [20.55] graph showing traffic variation in bits per second from 10AM to 4PM following day (example based on <http://oss.oetiker.ch/mrtg/>)

# ----- Report A -----					# ----- Report B -----				
#					#				
# Fields: Percent Total					# Fields: Total				
# Symbols: Enabled					# Symbols: Disabled				
# Sorting: Descending Field 2					# Sorting: None				
# Name: IP protocol					# Name: Overall Summary				
###					##				
# protocol flows octets packets					Total Flows : 15155				
#					Total Octets : 5491033				
udp 82.547 69.416 63.101					Total Packets : 38242				
gre 3.623 17.180 16.184					Total Time (1/1000 secs) (flows): 25687672				
tcp 9.066 10.345 14.217					Duration of data (realtime) : 86375				
icmp 4.764 3.059 6.498					Duration of data (1/1000 secs) : 86375588				
Packets per flow distribution:					Average flow time (1/1000 secs) : 1694.0000				
1	2	4	8	12	Average packet size (octets) : 143.0000				
.589	.072	.205	.110	.017	Average flow size (octets) : 362.0000				
					Average packets per flow : 2.0000				
					Average flows / second (flow) : 0.1755				
					Average flows / second (real) : 0.1755				
					Average Kbits / second (flow) : 0.5086				
					Average Kbits / second (real) : 0.5086				

Fig. 20.5 Flow-tools report showing various statistics extracted from NetFlow traces (example based on <http://www.singaren.net.sg/library/presentations/6nov02.pdf>)

on data sent/received, utilized bandwidth, IP multicast information, TCP sessions history, UDP traffic, TCP/UDP services used, and traffic distribution. Figure 20.7 shows a screendump of *ntop*'s global IP protocol distribution. In this particular network, it reveals the pre-dominance of UNIX based Network File System (NFS) transfers and X11 based X-Windows applications.

20.3.2 Monitoring Significant Aggregates of Traffic Volume

By looking at pre-defined coarse aggregates of traffic it is possible to miss many potentially important patterns. For example, if we look at the distribution of the number of packets per flow in Fig. 20.5, we can find that there are a large number of smaller flows (59% flows with 1 packet and 20% flows with 4 packets), than there are larger flows (11% flows with 8 packets and 1% flows with 12 packets). This shows that there may be significant patterns when some of these smaller patterns are aggregated. However, such patterns cannot always be identified by pre-defined coarse aggregates since all possible combinations of attributes and values are not considered.

A key issue in this context is how to define what is a "significant" aggregate flow. For example, in AutoFocus significant flows are combinations of uni-dimensional clusters whose traffic volume are above a given threshold. We discuss more about AutoFocus in Sect. 20.4. Similarly, Cormode et al. [20.28, 29] propose both offline and online techniques for aggregating traffic based on mining frequent items, known as *hierarchical heavy hitters*. Erman et al. [20.62] demonstrate the use of cluster based approaches to traffic classification. Kim et al. [20.30] use expert knowledge to construct characteristics of significant traffic patterns from flow statistics, in order to detect specific types of network attacks. Here the aggregate characteristics are matched against a table of possible patterns to identify an attack. For example, if a pattern contains a large number of flows but the ratio of packets/flow and flows/pattern is small, then it may be a scanning probe. On the other hand, if both the flow count and packet count are large and the destination is a broadcast address using the ICMP protocol, then it may be a *smurf* attack.

A key challenge in this context is how to efficiently search the space of all possible aggregate

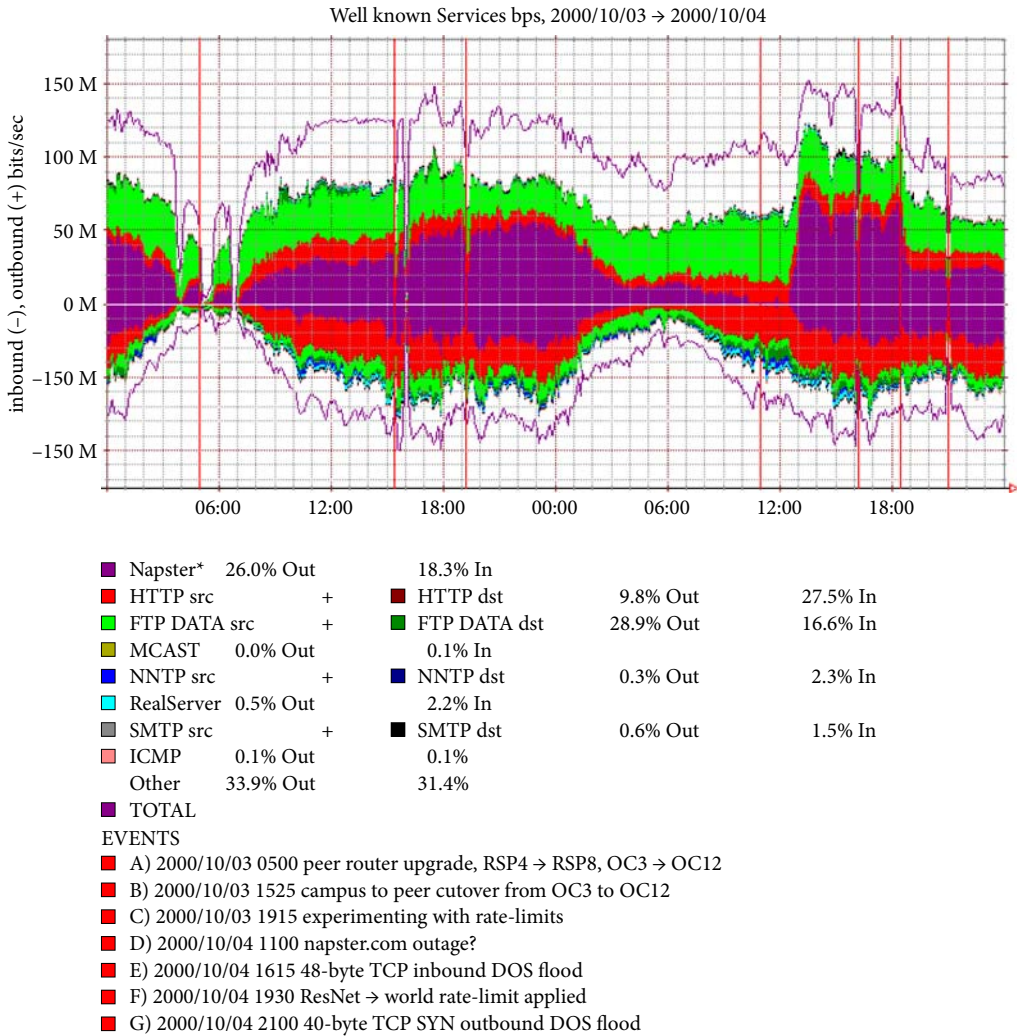


Fig. 20.6 Plonka’s FlowScan [20.58] tool showing the traffic snapshot of a campus over a time period. The port numbers and protocol information are used to infer the applications. For example, Napster uses a range of ports [6600–6699] for the clients and 8888 for the server communications. Napster is identified in *purple*. Another example is the FTP data, shown in *green*, which uses port 21 for commands and port 20 for transferring data using TCP (example based on <http://net.doit.wisc.edu/~plonka/lisa/FlowScan/>)

flows that could be monitored. In particular, we need to avoid storing all traffic records in memory or making multiple passes over the data. This general problem is the main focus of our research in this chapter. In Sect. 20.4, we present a case study of a relevant approach. First, let us consider one other family of approaches to analyzing traffic mixtures.

20.3.3 Monitoring Significant Changes in Traffic Volume

In network monitoring it is important to notice any significant changes in network traffic at an early stage. A significant rise in traffic volume may indicate a number of possible events, including a DoS attack, scan probe, traffic in peak hours, network-

Welcome
to
ntop!



About ntop
Data Rcvd
Data Sent
Stats

- o Multicast
- o Traffic
- o Hosts
- o Throughput
- o Domain
- o Plugins

 IP Traffic
IP Protocols
Admin

[No JavaScript]

Global IP Protocol Distribution

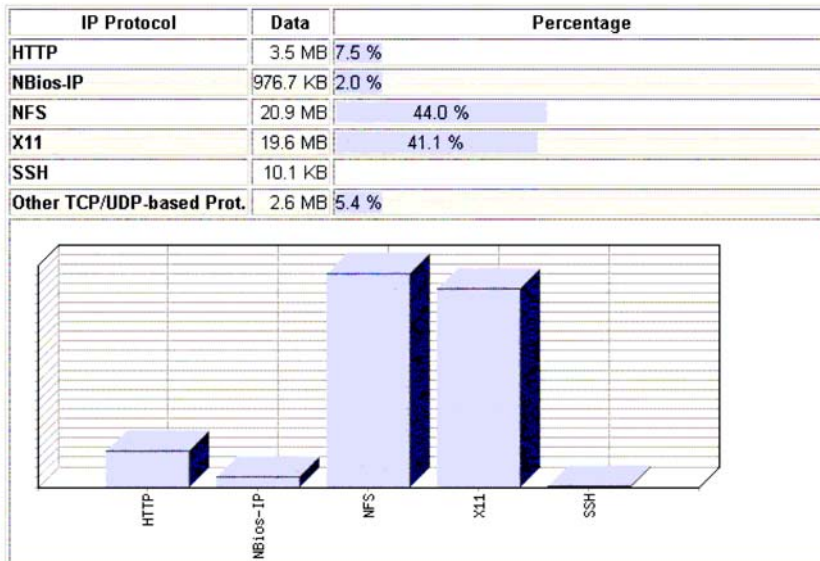


Fig. 20.7 A screendump showing ntop output in a browser window. The bar graph shows the relative usage of different IP protocols in the network (illustration has been taken from <http://www.simpleweb.org/tutorials/implementation/ntop/ntopa4.html>)

wide back up or file transfer traffic in off-peak hours. Conversely, a significant decline of traffic volume may also indicate that something may be wrong. For example, after a DoS attack or server compromise, some busy web or file server may not function properly or the server may have been rebooted which results in a decrease in traffic.

In particular, identifying significant changes in traffic clusters from two cluster reports requires finding clusters that are present in both reports such that their volume have changed significantly. Both Estan et al. [20.3] and Cormode et al. [20.63] suggest similar techniques for detecting changes in network traffic by computing *deltas* (or *deltoids*) from two snapshots of network traffic over time.

Finding changes is also important in fault detection [20.64–66]. For example, Feather et al. [20.67] detects faults by profiling normal traffic behavior and calculating statistical deviations from this normal behavior. Similar techniques have also been applied to the detection of intrusions and anomalies in network traffic [20.25, 30] by detecting changes from a normal model.

In this chapter, we do not consider the problem of finding significant changes, and we only mention it here for completeness. Our focus on the problem of finding significant aggregates, which can be a preliminary step to finding significant changes.

20.3.4 Frequency-Based Clustering Using Frequent Itemsets

As mentioned in Sect. 20.1.5, finding associations among different flows is important in traffic mixture analysis to identify groups of flows sharing a set of common characteristics. One way of finding associations is by generating all frequently occurring itemsets above a certain minimum value, where an itemset is simply a set of attribute values or items of a dataset.

Formally, consider a dataset D , which consists of a set of *transaction records* $D = \{X_1, \dots, X_N\}$. The structure of a record X depends on the type of application. In a transaction database containing records of customer purchases, a record X corresponds to a set of purchased *items*. Let $I = \{i_1, i_2, \dots, i_M\}$ de-

note the set of all items that can appear in records of the dataset D , e.g., the set of all products offered for sale. Then $D = \{X_j \mid j = 1, \dots, N, X_j \subseteq I\}$. In the case of a network trace file that contains a set of NetFlow records, the structure of a record is slightly different. Each record is a tuple of *attribute values* corresponding to a fixed set of attributes. Consider the set of attributes $A = \{\text{sourceIP}, \text{sourcePort}, \text{destinationIP}, \text{destinationPort}, \text{protocol}\}$. Then a record X corresponds to a tuple of values for these attributes, e.g., $X_1 = \{\text{sourceIP}_1, \text{sourcePort}_1, \text{destinationIP}_1, \text{destinationPort}_1, \text{protocol}_1\}$. In this case, we refer to each attribute value as an *item*.

An *itemset* corresponds to a set of items that appear in a dataset D . In our example of a dataset of NetFlow records, an itemset is a set of attribute values corresponding to a subset of the attributes in A . Let $C = \{\text{sourceIP}_1, \text{sourcePort}_1\}$. Then C can be used to represent all records in D with that combination of attribute values, i.e., all records with $\text{sourceIP} = \text{sourceIP}_1$ and $\text{sourcePort} = \text{sourcePort}_1$. If an attribute contains k items, then we refer to it as a k -*itemset*. The frequency of an itemset C is the number of records in D that contain the attribute values defined in C , e.g., the number of NetFlow records with $\text{sourceIP} = \text{sourceIP}_1$ and $\text{sourcePort} = \text{sourcePort}_1$. This is also known as the *support* of the itemset C with respect to a dataset D . A *frequent itemset* is one whose support is above a minimum threshold. Note that if C_1 is a frequent itemset, and $C_2 \subset C_1$, then C_2 is also a frequent itemset, since C_2 must match at least as many records as the more specific itemset C_1 .

Since an itemset provides a representation for a set of records in a dataset, it can be used as a representation for a cluster in frequency-based clustering (hence our use of the notation C for an itemset). Frequent itemset clustering involves finding all itemsets whose support is above a given threshold. Frequent itemset clustering on multi-dimensional data helps reveal information about the underlying usage patterns by combining the information derived from multiple attributes. The multi-dimensional clusters give an insight into the relations between different attributes. Manku et al. [20.68] identify different applications that apply frequent itemset calculation. For example, frequent itemsets are calculated in *iceberg* queries using group-by operators [20.69], for generating aggregates in OLAP data cube algorithms [20.70, 71], for finding association rules among frequent itemsets [20.72], and for finding

IP packet accounting information in network traffic measurement [20.13]. Next, we describe a frequent itemset clustering technique, AutoFocus, for network traffic data.

20.4 Case Study: AutoFocus

AutoFocus [20.3] is a tool for network traffic analysis, which uses frequent itemset mining to cluster network traffic flows. For each type of attribute in a network flow record, it first creates a uni-dimensional cluster tree of flows, and then combines these trees into a lattice structure to create a traffic report based on multidimensional clusters. For each of the uni-dimensional clustering and multi-dimensional clustering algorithms in AutoFocus, we describe the technique, illustrate the output with an example, and discuss the run-time and space complexity of the algorithm.

1. *Uni-dimensional clustering*: For each attribute, AutoFocus builds a one-dimension tree by counting frequent itemsets in the network traffic data [20.3]. This is straightforward for attributes such as protocols and ports. For protocols, the number of uni-dimensional itemsets is 2^8 and for ports, it is 2^{16} . However, the number of possible sets from the IP address space is much larger, i.e., 2^{32} . For IP addresses, it builds a tree of counters to reflect the structure of the IP address space. Counters at the leaves of the tree correspond to the original IP addresses that appeared in the traffic. In order to build an IP address prefix tree, AutoFocus goes through each record in the dataset to find the unique IP addresses and their corresponding count. Next, after arranging the leaf-counters in sorted order, it generates the prefix tree by computing the higher-level nodes corresponding to leaf-level IP addresses that have the same common prefix, i.e., addresses with the first l bits in common, where l is the level of the node in the tree. Since the total number of nodes in the tree is large, AutoFocus prunes the tree, by keeping only those nodes having traffic volumes above a threshold.
2. *Complexity of uni-dimensional clustering*: If m is the number of leaf nodes or unique IP addresses present in the tree and d is the depth of the tree, then the amount of memory required by this algorithm is $O(1 + m(d - 1))$. The running time

of the algorithm is $O(n + 1 + m(d - 1))$, where n is the number of records to be clustered.

3. *Example of uni-dimensional clusters:* Table 20.8 gives an example of a network traffic flow report. The first field gives information on the protocol used for communication. The UDP and ICMP protocols end with “/u” and “/i” after the protocol name or value. TCP protocols are only identified with their names and do not contain any “/”. The second and third fields are the source and destination ports used for a particular protocol. The fourth and fifth fields are the source and destination IP addresses of each flow. The sixth field mentions how many packets were involved in this flow. The example traffic data from Table 20.8 was used to generate an output from uni-dimensional clustering using the AutoFocus tool. Tables 20.3–20.7 show the uni-dimensional cluster reports generated by AutoFocus on this data. Table 20.3 shows the protocol breakdown of the total traffic. In this case AutoFocus has used protocol numbers in the protocol field instead of their names. The Internet Assigned Numbers Authority (IANA) is the central coordinator for the assignment of the values for Internet protocols. The list of all assigned protocol value and name pairs can be found in [20.73]. Protocol value 1 is assigned to ICMP, protocol 6 is assigned to TCP and protocol 17 is assigned to UDP. In the example dataset most of the reported traffic belongs to ICMP, followed by TCP and only a few UDP packets. Tables 20.4 and 20.5 show the traffic by source and destination IP addresses. Similarly, Tables 20.6 and 20.7 show the traffic by source and destination ports. Such uni-dimensional breakdowns are also common in other network traffic reporting tools, such as MRTG, and may help identify the IP addresses or applications having a greater influence on the bandwidth than the rest.
4. *Multidimensional clustering:* For multidimensional clustering, AutoFocus uses the combination of m uni-dimensional cluster trees to create an m -dimensional lattice structure. For example, the top right part of Fig. 20.8 shows a prefix tree, which shows the break up of traffic originating from various departments in a university (shown as E and M), and the top left part shows a protocol tree, which shows the traffic

Table 20.3 AutoFocus Protocol report

Protocol	Breakdown	
	Percentage	# records
1	51.79%	29
6	42.86%	24
17	5.36%	3

Table 20.4 AutoFocus Source IP report

Source IP	Breakdown	
	Percentage	# records
172.16.112.20/32	3.57%	2
172.16.114.148/32	42.86%	24
199.174.194.0/24	32.14%	18
199.174.194.0/27	7.14%	4
199.174.194.6/31	3.57%	2
199.174.194.64/26	7.14%	4
199.174.194.64/28	3.57%	2
199.174.194.128/27	3.57%	2
199.174.194.160/27	3.57%	2
199.174.194.220/30	3.57%	2
199.174.194.224/27	3.57%	2
208.240.124.83/32	19.64%	11

Table 20.5 AutoFocus Destination IP report

Destination IP	Breakdown	
	Percentage	# records
172.16.112.0/27	21.43%	12
172.16.112.2/31	3.57%	2
172.16.112.4/31	3.57%	2
172.16.112.6/31	3.57%	2
172.16.112.8/31	3.57%	2
172.16.112.10/31	3.57%	2
172.16.114.50/32	32.14%	18
192.168.1.0/27	3.57%	2
199.95.74.90/32	42.86%	24

Table 20.6 AutoFocus Source Port report

Source port	Breakdown	
	Percentage	# records
53	3.57%	2
1024–65,535	42.86%	24
1173	3.57%	2

Table 20.7 AutoFocus Destination Port report

Destination port	Breakdown	
	Percentage	# records
0–1023	46.43%	26
80	42.86%	24

Table 20.8 Example of a network traffic flow report

Flow#	Protocol	Src. Port	Dst. Port	Source IP	Destination IP	Pkts
1	ecr/i	-	-	199.174.194.086	172.016.114.050	1
2	ecr/i	-	-	199.174.194.159	172.016.114.050	1
3	ecr/i	-	-	199.174.194.204	172.016.114.050	1
4	ecr/i	-	-	199.174.194.172	172.016.114.050	1
5	ecr/i	-	-	199.174.194.076	172.016.114.050	1
6	ecr/i	-	-	199.174.194.007	172.016.114.050	1
7	ecr/i	-	-	199.174.194.251	172.016.114.050	1
8	ecr/i	-	-	199.174.194.102	172.016.114.050	1
9	ecr/i	-	-	199.174.194.011	172.016.114.050	1
10	ecr/i	-	-	199.174.194.017	172.016.114.050	1
11	ecr/i	-	-	199.174.194.006	172.016.114.050	1
12	ecr/i	-	-	199.174.194.136	172.016.114.050	1
13	ecr/i	-	-	199.174.194.221	172.016.114.050	1
14	ecr/i	-	-	199.174.194.050	172.016.114.050	1
15	ecr/i	-	-	199.174.194.191	172.016.114.050	1
16	ecr/i	-	-	199.174.194.222	172.016.114.050	1
17	ecr/i	-	-	199.174.194.227	172.016.114.050	1
18	ecr/i	-	-	199.174.194.067	172.016.114.050	1
19	eco/i	-	-	208.240.124.083	172.016.112.001	1
20	eco/i	-	-	208.240.124.083	172.016.112.002	1
21	eco/i	-	-	208.240.124.083	172.016.112.003	1
22	eco/i	-	-	208.240.124.083	172.016.112.004	1
23	eco/i	-	-	208.240.124.083	172.016.112.005	1
24	eco/i	-	-	208.240.124.083	172.016.112.006	1
25	eco/i	-	-	208.240.124.083	172.016.112.007	1
26	eco/i	-	-	208.240.124.083	172.016.112.008	1
27	eco/i	-	-	208.240.124.083	172.016.112.009	1
28	eco/i	-	-	208.240.124.083	172.016.112.010	1
29	eco/i	-	-	208.240.124.083	172.016.112.011	1
30	http	1026	80	172.016.114.148	199.095.074.090	24
31	ntp/u	123	123	172.016.112.020	192.168.001.010	1
32	domain/u	53	1233	192.168.001.010	172.016.112.020	1
33	domain/u	53	53	172.016.112.020	192.168.001.020	1

belonging to the TCP and UDP protocols. These two uni-dimensional trees are then combined to build the multi-dimensional structure in the bottom part of Fig. 20.8. By doing a top-down level-wise traversal with each uni-dimensional tree, the algorithm combines nodes from one tree with the nodes from the other tree. For example, combining E from the prefix tree with T and U from the protocol tree produces the children TE and UE which represent TCP and UDP traffic from the Engineering department. Furthermore, C and UE are combined to produce their child UC, which represents the UDP traffic originating from the Computer Science department.

5. *Complexity of multi-dimensional clustering:* As mentioned before, in order to create multi-

dimensional clusters it is first necessary to create the uni-dimensional trees, which is $O(n + 1 + m(d - 1))$. In order to create the multi-dimensional structure, the combination steps require looking through approximately $n \prod_{i=1}^m d_i$ itemsets, which is the product of the depth of each of the uni-dimensional trees and the number of input flows [20.3]. Building the complete lattice would be expensive since it involves all possible combinations among the values of different attributes in the worst case. Instead, AutoFocus uses certain properties of the lattice structure to avoid brute force enumeration. Nevertheless, AutoFocus still requires multiple passes through the network traffic dataset in order to generate frequent multidimensional clusters. The memory re-

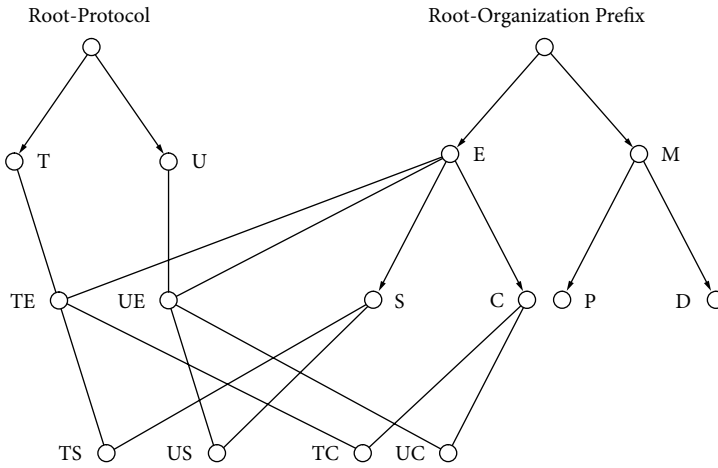


Fig. 20.8 Example of a multi-dimensional cluster lattice shows how two uni-dimensional clusters trees based on protocol and department prefix are combined to build a lattice structure of multi-dimensional clusters. T and U are TCP and UDC protocols. E and M are fictitious Engineering and Medical faculties of a university. S and C are the Statistics and Computer Science Departments. P and D are Paediatric and Dermatology Departments. TE, UE and UC are combined traffic clusters from TCP-Engineering, UDP-Engineering and UDP-Computer Science

quirement to store the candidate clusters in memory is also high and even with optimization is in the order of $s \prod_{i=1}^m d_i$, where $s = n/h$, s is the number of the large clusters that will be reported for a threshold h and total input records n . The following example will help us understand how AutoFocus generates its multi-dimensional clusters and highlight some of its shortcomings in terms of the large size of report as well as not being able to find many important clusters.

6. *Example of multi-dimensional clusters:* In the context of network traffic analysis, it is more important to look at a combination of the uni-dimensional fields to better understand any underlying patterns. Table 20.9 shows a multi-dimensional report generated by AutoFocus from the same example data in Table 20.8. The traffic report corresponds to network clusters generated by AutoFocus and it lists the more general clusters before the more specific ones. For example, the first line of the report tells

Table 20.9 Example of a multi-dimensional report from AutoFocus

Entry	Source IP	Destination IP	Pro	Src Port	Dst Port	Pkts
1	*	*	17	53	*	2
2	*	172.16.112.0/27	*	*	*	12
3	172.16.112.20/32	192.168.1.0/27	17	0-1023	0-1023	2
4	172.16.114.148/32	199.95.74.90/32	6	1024-65,535	80	24
5	172.16.114.148/32	199.95.74.90/32	6	1173	80	2
6	199.174.194.0/24	172.16.114.50/32	1	*	*	18
7	199.174.194.0/27	172.16.114.50/32	1	*	*	4
8	199.174.194.6/31	172.16.114.50/32	1	*	*	2
9	199.174.194.64/26	172.16.114.50/32	1	*	*	4
10	199.174.194.64/28	172.16.114.50/32	1	*	*	2
11	199.174.194.128/27	172.16.114.50/32	1	*	*	2
12	199.174.194.160/27	172.16.114.50/32	1	*	*	2
13	199.174.194.220/30	172.16.114.50/32	1	*	*	2
14	199.174.194.224/27	172.16.114.50/32	1	*	*	2
15	208.240.124.83/32	172.16.112.2/31	1	*	*	2
16	208.240.124.83/32	172.16.112.4/31	1	*	*	2
17	208.240.124.83/32	172.16.112.6/31	1	*	*	2
18	208.240.124.83/32	172.16.112.8/31	1	*	*	2
19	208.240.124.83/32	172.16.112.10/31	1	*	*	2

us that there are just two packets that belong to the UDP (protocol 17) and use source port 53. Similarly, the second line indicates that the destination IP addresses 172.16.112.0/27 are the recipient of 12 packets.

20.5 How Can We Apply Network Traffic Monitoring Techniques for SCADA System Security?

20.5.1 SCADA Systems

SCADA (Supervisory Control and Data Acquisition) systems are computer based tools to control and monitor industrial and critical infrastructure functions, such as the generation, transmission and distribution of electricity, gas, water, waste, railway and traffic control in real time. All of these utilities are essential in the proper functioning of our daily life, therefore its security and protection are extremely important as well as of national concern.

The primary function of a SCADA system is to efficiently connect and transfer information from a wide range of sources, and at the same time maintaining data integrity and security.

SCADA systems have been around since the 1960s, when the direct human involvement in monitoring and control of utility plants was gradually replaced by remote operation of valves and switches through the use of modern telecommunication devices such as phones lines and dedicated circuits. The emergence of powerful personal computers and servers and the need to connect to the Internet have added a new dimension to the operation of SCADA systems. For example, the operator can remotely login to the SCADA systems without the need to be physically present at the remote control sites. Unfortunately, this has also led to an opportunity for intruders and attackers to compromise the system by posing as a legitimate operator or by taking control of the operator's computer.

Figure 20.9 illustrates how a modern SCADA system is connected. The field devices consist of Remote Terminal Units (RTU), Programmable Logic Devices (PLC), and Intelligent Electronic Devices (IED). A number of RTUs in remote locations collect data from devices and send log data and alarms to a SCADA terminal using various communication links including traditional telephone and computer

network, wireless network, and fiber optic cables. Data acquisition begins at the RTU or PLC level and includes meter readings and equipment status reports that are communicated to SCADA as required. Some industrial systems use PLCs to control end devices like sensors and actuators. Data from the RTUs and PLCs is compiled and formatted in such a way that a control room operator using a Human Machine Interface (HMI) can make supervisory decisions to adjust or override normal RTU (or PLC) controls. This data may also be collected and stored in a *Historian*, a type of Database Management System, to allow auditing, and the analysis of trends and anomalies.

20.5.2 SCADA Security Issues

Today many of the SCADA systems are also connected to the corporate network where a manager or an engineer can view and change control settings. The data is transferred through a communication server that is protected by a firewall from the corporate network which is often connected to the wider Internet. The SCADA data is increasingly being transported using the TCP/IP protocol for increased efficiency, enhance interconnectivity, and because of the ease of using commercial-off-the-shelf hardware and software. Protocols such as Modbus and DNP3 that had been traditionally used for interconnection within SCADA network are increasingly being transported over TCP/IP as the field devices are also providing IP support [20.75]. This leads to a standardized and transparent communication model both within and outside the SCADA network. As TCP/IP is becoming the predominant carrier protocol in modern SCADA networks, it introduces the potential for innovative attacks targeting the SCADA system, which had been previously isolated from the corporate information technology and communications infrastructure. Since most SCADA protocols were not designed with security issues in mind, therefore, an attack on the TCP/IP carrier could expose the unprotected SCADA data. In addition, traditional attacks from the Internet could be transported through the interconnected corporate network into the SCADA network and disrupt the industrial processes [20.76, 77]. The various network monitoring functions can help protect a SCADA network by continuously monitoring incoming and outgoing

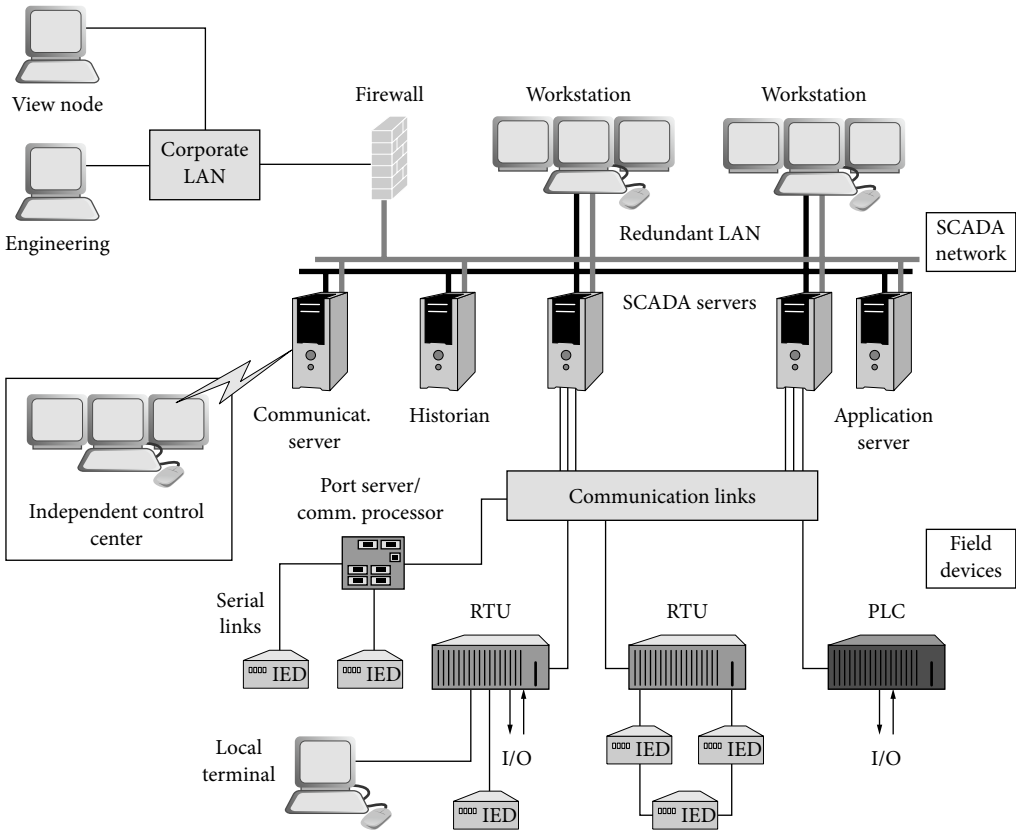


Fig. 20.9 An illustration of a SCADA system showing how the SCADA servers are connected to both the field devices and the corporate LAN. Example taken from [20.74]

traffic from the SCADA network, and by generating alarms in an accurate and efficient manner for real time response. A general architecture for monitoring traffic at different parts of the SCADA network is discussed below.

20.5.3 Protecting SCADA Systems by Using Network Traffic Monitoring

As shown in Fig. 20.10, SCADA system is different from normal TCP/IP network. In addition to the normal TCP/IP network, a SCADA system has its own industrial process which is normally involving industrial specific networking protocols. No literature report has been found on how to use network traffic monitoring management for the protection of

the SCADA systems. In this chapter, an architecture of network traffic monitoring management is suggested as shown in Fig. 20.10 for the protection of the SCADA systems.

This is a distributed network traffic monitoring architecture. In this architecture, monitoring sensors A,B,C, and D are deployed in the system. Monitor A is deployed between the Corporate LAN and the firewall of the SCADA network. Monitor B is deployed immediately after the firewall of the SCADA network. This arrangement can monitor the network traffic attempting to access the SCADA system and network traffic that has eventually gone through the firewall. As new attacks can potentially penetrate the firewall, it is essential to monitor all traffic that has successfully passed the firewall.

Monitor C is monitoring all traffic flowing within the SCADA LAN. Monitor D is placed between the

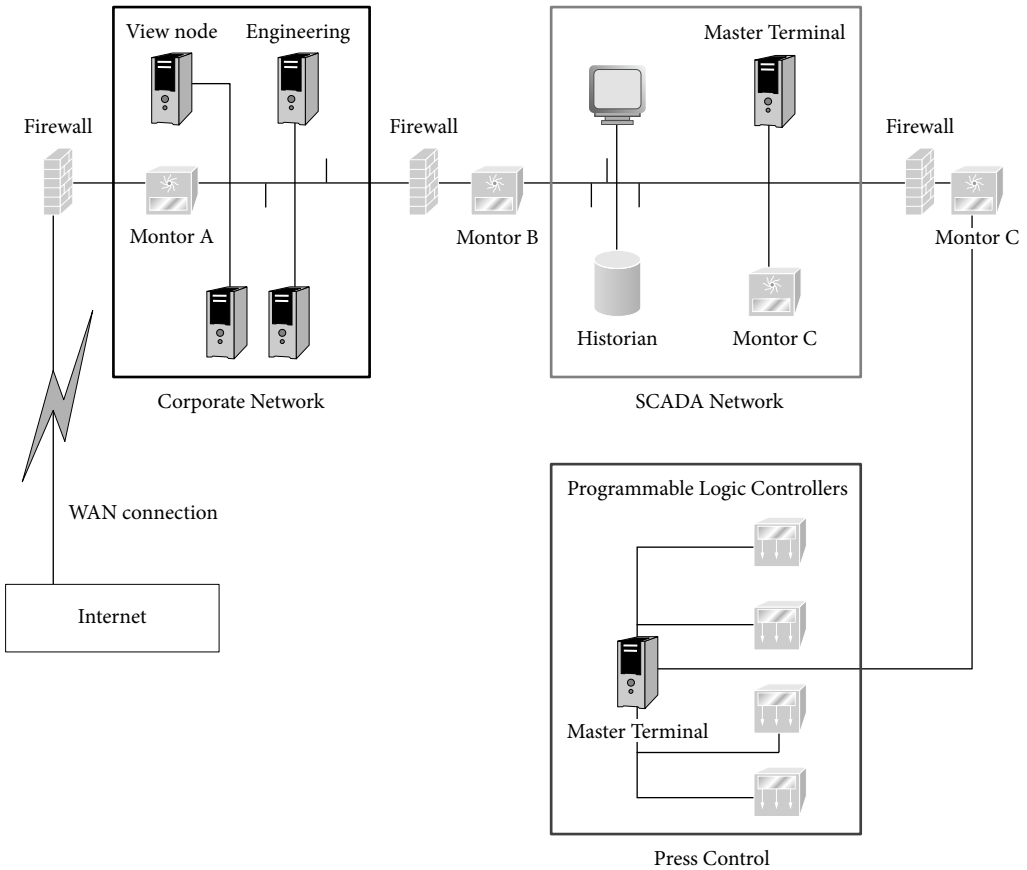


Fig. 20.10 Monitoring a SCADA network

SCADA server and the field devices. It can monitor specific industrial protocol traffic. It is preferable to use passive monitoring techniques to minimize the potential risk induced by active probing.

Some process controls in SCADA networks experience bursty traffic, therefore it is appropriate to apply frequent itemset traffic analysis to monitor any unusual traffic in the network. The AutoFocus tool introduced in Sect. 20.4 would be a useful tool to monitor end to end flows.

20.6 Conclusion

A fundamental problem in the management of IP networks including critical SCADA networks is how to analyze network traffic to identify significant patterns of network usage. In this chapter,

we have summarized the general types of traffic analysis problems that arise in this context, and highlighted our focus on traffic volume and traffic mixture analysis. We then describe the relevant methods for collecting raw traffic measurements, and the related work on analyzing the mixture of traffic in these measurements. In particular, our focus is on identifying significant aggregates by volume given a trace of network flow records. We have described an existing approach to this problem in detail, namely, frequent itemset clustering using an illustration for a case study based on AutoFocus.

Acknowledgements This work is partially supported by ARC (Australia Research Council) Discovery Grant DP0985838.

References

- 20.1. M. Sloman: *Network and Distributed Systems Management* (Addison-Wesley Longman, Boston, MA, USA 1994)
- 20.2. A. Mahmood, C. Leckie, P. Udaya: An efficient clustering scheme to exploit hierarchical data in network traffic analysis, *IEEE Trans. Knowl. Data Eng.* **20**(6), 752–767 (2008)
- 20.3. C. Estan, S. Savage, G. Varghese: Automatically inferring patterns of resource consumption in network traffic, *Proc. ACM SIGCOMM Conference* (2003)
- 20.4. S. Keshav: An engineering approach to computer networking: ATM networks, the internet, and the telephone network (Addison-Wesley Longman, Boston, MA, USA 1997)
- 20.5. Z. Dziong, J. Roberts: Congestion probabilities in a circuit-switched integrated services network, *Perform. Eval.* **7**(4), 267–284 (1987)
- 20.6. F. Kelly: Routing in circuit-switched networks: optimization, shadow prices and decentralization, *Adv. Appl. Probab.* **20**(1), 112–144 (1988)
- 20.7. Y. Vardi: Network tomography: estimating source-destination traffic intensities from link data, *J. Am. Stat. Assoc.* **91**(433), 365–377 (1996)
- 20.8. A. Medina et al.: Traffic matrix estimation: existing techniques and new directions, *Proc. 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (2002) pp. 161–174
- 20.9. J. Cao: Time-varying network tomography: router link data, *J. Am. Stat. Assoc.* **95**(452), 1063–1075 (2000)
- 20.10. O. Goldschmidt: ISP backbone traffic inference methods to support traffic engineering, *Internet Statistics and Metrics Analysis (ISMA) Workshop* (2000) pp. 1063–1075
- 20.11. C. Tebaldi, M. West: Bayesian inference of network traffic using link count data, *J. Am. Stat. Association* **93**(442), 557–573 (1998)
- 20.12. M. Cai et al.: Fast and accurate traffic matrix measurement using adaptive cardinality counting, *Applications, Technologies, Architectures, and Protocols for Computer Communication* (2005) pp. 205/206
- 20.13. C. Estan, G. Varghese: New directions in traffic measurement and accounting: focusing on the elephants, ignoring the mice, *ACM Trans. Comput. Syst.* **21**(3), 270–313 (2003)
- 20.14. B. Roh, S. Yoo: A novel detection methodology of network. In: *Proc. ICT, Intl. Conf. on Telecom.* Vol. 3124, (Springer Berlin, Heidelberg 2004) pp. 1226–1235
- 20.15. Network monitoring tools, available at <http://www.slac.stanford.edu/xorg/nmtf/nmtf-tools.html>
- 20.16. Network visualization tools, available at <http://www.caida.org/funding/internetatlas/viz/viztools.html> (2000)
- 20.17. Flow-tools, available at <http://www.splintered.net/sw/flow-tools/> (2000)
- 20.18. cflowd: Traffic flow analysis tool, available at <http://www.caida.org/tools/measurement/cflowd/> (2000)
- 20.19. R. Addie, M. Zukerman, T. Neame: Broadband traffic modeling: simple solutions to hard problems, *Commun. Mag. IEEE* **36**(8), 88–95 (1998)
- 20.20. W. Willinger, V. Paxson: Where mathematics meets the internet, *Notices Am. Math. Soc.* **45**(8), 961–970 (1998)
- 20.21. J. Bolot: Characterizing end-to-end packet delay and loss in the Internet, *J. High-Speed Netw.* **2**(3), 305–323 (1993)
- 20.22. P. Huang, A. Feldmann, W. Willinger: A non-intrusive, wavelet-based approach to detecting network performance problems, *Internet Measurement Workshop* (2005)
- 20.23. Y. Zhang, N. Duffield: On the constancy of internet path properties, *Proc. 1st ACM SIGCOMM Workshop on Internet Measurement* (2001) pp. 197–211
- 20.24. V. Paxson: End-to-end internet packet dynamics, *IEEE/ACM Trans. Netw.* **7**(3), 277–292 (1999)
- 20.25. P. Barford et al.: A signal analysis of network traffic anomalies, *Proc. 2nd ACM SIGCOMM Workshop on Internet Measurement* (ACM Press, New York, NY, USA 2002)
- 20.26. S. Kim, A. Reddy, M. Vannucci: Detecting traffic anomalies through aggregate analysis of packet header data. In: *Networking'2004, Lecture Notes in Computer Science*, Vol. 3042, ed. by N. Mitrou, K. Kontovasilis, G.N. Rouskas, I. Iliadis, L. Merakos (Springer, Berlin Heidelberg 2004) pp. 1047–1059
- 20.27. AutoFocus tool, available at <http://www.caida.org/tools/measurement/autofocus/> (2003)
- 20.28. G. Cormode et al.: Finding hierarchical heavy hitters in data streams, *Proc. VLDB* (2003)
- 20.29. G. Cormode et al.: Diamond in the rough: finding hierarchical heavy hitters in multi-dimensional data, *Proc. ACM SIGMOD* (ACM Press, New York, NY, USA 2004)
- 20.30. M. Kim et al.: A flow-based method for abnormal network traffic detection, *IEEE/IFIP Network Operations and Management Symposium*, Seoul (2004)
- 20.31. P. Chhabra, A. John, H. Saran: PISA: automatic extraction of traffic signatures. In: *Networking'2005, Lecture Notes in Computer Science*, Vol. 3462, ed. by R. Boutaba, K. Almeroth, R. Puigianer, S. Shen, J.P. Black (Springer, Berlin Heidelberg 2005) pp. 730–742
- 20.32. N. Duffield, C. Lund, M. Thorup: Charging from sampled network usage, *Proc. 1st ACM SIGCOMM Workshop on Internet Measurement* (ACM Press, New York, NY, USA 2001)

- 20.33. K. Claffy, G. Polyzos, H. Braun: Application of sampling methodologies to network traffic characterization, *ACM SIGCOMM Comput. Commun. Rev.* **23**(4), 194–203 (1993)
- 20.34. A. Kumar et al.: Data streaming algorithms for efficient and accurate estimation of flow size distribution, *Proc. ACM SIGMETRICS/Performance* (2004)
- 20.35. N. Alon, Y. Matias, M. Szegedy: The space complexity of approximating the frequency moments, *J. Comput. Syst. Sci.* **58**(1), 137–147 (1999)
- 20.36. P. Benko, A. Veres: A passive method for estimating end-to-end tcp packet loss, *Global Telecommunications Conference* (2002)
- 20.37. S. Jaiswal et al.: Inferring TCP connection characteristics through passive measurements, *INFOCOM 2004, 23rd Annual Joint Conference of the IEEE Computer and Communications Societies* (2004)
- 20.38. S. Jaiswal: Measurement and classification of out-of-sequence packets in a Tier-1 IP backbone, *IEEE/ACM Trans. Netw.* **15**(1), 54–66 (2007)
- 20.39. H. Jiang, C. Dovrolis: Passive estimation of TCP round-trip times, *ACM SIGCOMM Comput. Commun. Rev.* **32**(3), 75–88 (2002)
- 20.40. S. Katti et al.: M&M: A passive toolkit for measuring, tracking and correlating path characteristics, *Proc. ACM Internet Measurements Conference* (2004)
- 20.41. Y. Zhang et al.: On the characteristics and origins of internet flow rates, *Proc. 2002 SIGCOMM Conference* (ACM Press, New York, NY, USA 2002)
- 20.42. J. Curtis: Principles of passive measurement, available at <http://www.wand.net.nz/pubs/19/html/node10.html> (2007)
- 20.43. M. Muuss: The story of the PING program, available at <http://ftp.arl.mil/mike/ping.html> (1983)
- 20.44. G. Malkin: Traceroute using an IP option, RFC1393 (January, 1993), available at <http://tools.ietf.org/html/rfc1393>
- 20.45. V. Jacobson: Pathchar-A tool to infer characteristics of internet paths, available at <http://tools.ietf.org/html/rfc1393> (1997)
- 20.46. K. Kendall: A database of computer attacks for the evaluation of intrusion detection systems, M.Sc. Thesis (MIT Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1999) p. 124
- 20.47. J. Cleary et al.: Design principles for accurate passive measurement, *Proc. Passive and Active Measurement Workshop* (2000)
- 20.48. WAND Network Research Group: <http://www.wand.net.nz/> (2007)
- 20.49. Endace network monitoring, latency measurement and application acceleration solutions, available at <http://www.endace.com/> (2007)
- 20.50. J. Kurose, K. Ross: *Computer Networking* (Addison-Wesley, Boston 2003)
- 20.51. Cisco: Introduction to Cisco IOS NetFlow – a technical overview (Cisco Systems Inc., 2007)
- 20.52. A. Myers: JFlow: Practical mostly-static information flow control, *Proc. 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages* (ACM Press, New York, NY, USA 1999)
- 20.53. Technical white paper for NetStream, available at <http://www.huawei.com/products/datacomm/pdf/view.do?f=65> (2007)
- 20.54. B. Claise, M. Fullmer: IPFIX protocol specifications, Draft IETF IPFIX Protocol-3 (2004)
- 20.55. T. Oetiker: MRTG – the multi router traffic grapher, *Proc. 12th Systems Administration Conference (LISA'98)* (1998)
- 20.56. Cricket: A high performance system for monitoring trends in time-series data, available at <http://cricket.sourceforge.net/> (2003)
- 20.57. T. Oetiker: The RRDtool manual, available at <http://ee-staff.ethz.ch/oetiker/webtools/rrdtool/manual/index.html> (1998)
- 20.58. D. Plonka: FlowScan: a network traffic flow reporting and visualization tool, 14th USENIX Conference on System Administration, New Orleans, LA (2000)
- 20.59. S. Leinen: Fluxoscope: a system for flow-based accounting, available at <http://www.tik.ee.ethz.ch/cati/deliv/CATI-SWI-IM-P-000-0.4.pdf> (2000)
- 20.60. L. Deri, S. Suin: Ntop: beyond ping and traceroute. In: *Active Technologies for Network and Service Management*, Lecture Notes in Computer Science, Vol. 1700, ed. by R. Stadler, B. Stiller (Springer, Berlin Heidelberg 1999) pp. 271–283
- 20.61. S. Romig, M. Fullmer, R. Luman: The OSU flow-tools package and Cisco NetFlow logs, *Proc. USENIX LISA* (1995)
- 20.62. J. Erman, M. Arlitt, A. Mahanti: Traffic classification using clustering algorithms, *Proc. ACM SIGCOMM Workshop on Mining Network Data (MineNet)*, Pisa, Italy (2006)
- 20.63. G. Cormode, S. Muthukrishnan: What's new: finding significant differences in network data streams, *IEEE/ACM Trans. Netw.* **13**(6), 1219–1232 (2005)
- 20.64. C. Hood, C. Ji: Proactive network fault detection, *Proc. IEEE INFOCOM'97*, Kobe, Japan (1997)
- 20.65. I. Katzela, M. Schwartz: Schemes for fault identification in communications networks, *IEEE/ACM Trans. Netw.* **3**(6), 753–764 (0000)
- 20.66. A. Ward, P. Glynn, K. Richardson: Internet service performance failure detection, *Proc. Internet Server Performance Workshop* (1998)
- 20.67. F. Feather, D. Siewiorek, R. Maxion: Fault detection in an ethernet network using anomaly signature matching, *Applications, Technologies, Architectures, and Protocols for Computer Communication* (1993) pp. 279–288
- 20.68. G. Manku, R. Motwani: Approximate frequency counts over data streams, *Proc. 28th International*

- Conference on Very Large Data Bases (VLDB 2002) (Morgan Kaufmann, 2002)
- 20.69. M. Fang et al.: Computing iceberg queries efficiently, Proc. 24th International Conference on Very Large Data Bases (VLDB 1998) (1998)
- 20.70. J. Han, M. Kamber: *Data Mining: Concepts and Techniques* (Morgan Kaufmann, San Francisco 2006) p. 550
- 20.71. K. Beyer, R. Ramakrishnan: Bottom-up computation of sparse and iceberg CUBE, ACM SIGMOD Rec. **28**(2), 359–370 (1999)
- 20.72. R. Agrawal, R. Srikant: Fast algorithms for mining association rules, Proc. VLDB 1994, 20th International Conference of Very Large Data Bases (1994)
- 20.73. Internet Assigned Numbers Authority (2008)
- 20.74. Pacific Northwest National Laboratory: The role of authenticated communications for electric power distribution, Position Paper for the National Workshop – Beyond SCADA: Networked Embedded Control for Cyber Physical Systems (Pacific Northwest National Laboratory, U.S. Department of Energy, 2006)
- 20.75. R. Chandia, J. Gonzalez, T. Kilpatrick, M. Papa, S. Shenoi: *Critical Infrastructure Protection (IFIP International Federation for Information Processing)*, ed. by S. Shenoi (Springer, Boston 2008) pp. 117–131
- 20.76. M. Berg, J. Stamp: A reference model for control and automation systems in electric power, Sandia National Laboratories, available at http://www.sandia.gov/scada/documents/sand_2005_1000C.pdf
- 20.77. E. Byres et al.: Worlds in collision: ethernet on the plant floor, Proc. ISA Emerging Technologies Conference, Instrumentation Systems and Automation Society (2002)

The Authors



Abdun Naser Mahmood received the BSc degree in Applied Physics and Electronics and the MSc degree in Computer Science from the University of Dhaka, Bangladesh, in 1997 and 1999, respectively. He completed his PhD degree from the University of Melbourne in 2008. He joined the University of Dhaka as a lecturer in 2000, Assistant Professor in 2003, when he took a leave of absence for his PhD studies. Currently, he is a Postdoctoral Research Fellow at the Royal Melbourne Institute of Technology with the School of Computer Science and Information Technology. His research interests include data mining techniques for network monitoring and algorithm design for adaptive sorting and sampling.

Abdun Mahmood
School of Computer Science and IT
RMIT University
Melbourne 3001, Australia
abdun.mahmood@cs.rmit.edu.au



Dr Christopher Leckie is an Associate Professor in the Department of Computer Science and Software Engineering at the University of Melbourne, Australia. He has made numerous theoretical contributions to the use of clustering for problems such as anomaly detection in wireless sensor networks and the Internet. In particular, he has developed efficient clustering techniques that are specifically designed to cope with high-dimensional and time-varying data streams, which are a major challenge in network intrusion detection. His work on filtering denial-of-service attacks on the Internet has been commercialized with an Australian company, leading to a commercial product. His research has been published in leading journals and conferences such as ACM Computing Surveys, IEEE TKDE, Artificial Intelligence, IJCAI and ICML.

Christopher Leckie
Department of Computer Science & Software Engineering
The University of Melbourne
Melbourne 3052, Australia
caleckie@csse.unimelb.edu.au



Jiankun Hu obtained his Masters Degree from Department of Computer Science and Software Engineering of Monash University, Australia; PhD degree from Control Engineering, Harbin Institute of Technology, China. He has been awarded the German Alexander von Humboldt Fellowship working at Ruhr University, German. He is currently an Associate Professor at the School of Computer Science and IT, RMIT University. He leads the Networking Cluster within the Discipline of Distributed Systems and Networks. Dr. Hu's current research interests are in network security with emphasis on biometric security, mobile template protection and anomaly intrusion detection. These research activities have been funded by three Australia Research Council (ARC) Grants. His research work has been published on top international journals.

Jiankun Hu
School of Computer Science and IT
RMIT University
Melbourne 3001, Australia
jiankun@cs.rmit.edu.au



Zahir Tari is a full professor at RMIT University. He is also the Director/Leader of the DSN (Distributed Systems & Networking) discipline at the School of Computer Science & IT, RMIT (Australia). His research interests are mainly in the area of performance (e.g. WEB SERVERS, CDN, P2P), security (e.g. access control, information flow control, inference) and web services in general (e.g. service matching, verification of communication protocols). He acted as the program committee chair as well as general chair of more than 16 international conferences. Recently, Professor Tari has been leading a research initiative in the area of SCADA security, which is supported by the iPlatform Institute within RMIT University. The focus of the research group is on the development of both theoretical framework (for the various security aspects, including IDS and survivability) as well as specific testbed.

Zahir Tari
School of Computer Science and IT
RMIT University
Melbourne 3001, Australia
zahir.tari@cs.rmit.edu.au



Mohammed Atiquzzaman obtained his MS and PhD in Electrical Engineering and Electronics from the University of Manchester (UK). He is currently a Professor in the School of Computer Science at the University of Oklahoma, and a senior member of IEEE. Dr. Atiquzzaman is the E-i-C of Journal of Networks and Computer Applications, co-E-i-C of Computer Communications Journal. He is the co-author of the book *Performance of TCP/IP over ATM Network* and has over 150 refereed publications. His current research interests are in areas of transport protocols, wireless and mobile networks, ad hoc networks, satellite networks, quality of service, and optical communications. His research has been funded by National Science Foundation (NSF), National Aeronautics and Space Administration (NASA), and U.S. Air Force.

Mohammed Atiquzzaman
School of Computer Science
University of Oklahoma
Norman, OK 73019-6151, USA
atiq@ou.edu