Network Traffic Inference Using Sampled Statistics

Hamed Haddadi
University College London

Supervisor: Dr Miguel Rio

August 1, 2006

Abstract

This report aims to summarise the current research trends and challenges of monitoring high speed networks. The report also presents the work carried out by the author in this field, the tools which have been made available to the community by the author and the future directions of this research project work. A summary of the developed simulation test bed and its application in the research project context is also discussed.

The emergence of research in e-Science contexts ranging from materials simulation to physics measurements has lead to a rapid increase of process-intensive and bandwidth-hungry applications which need to use several Giga bits per second flows and increasingly higher data storage volumes. In the UK e-Science community, the UKLight high-speed switched network [1] is set up with dedicated 10Gbps links to international research networks. This facility enables network researchers to try all kinds of application at network layer, transport layer and session layer, ranging from variations of TCP implementations to Grid computing file transfer protocols.

This report covers some of the challenges in measurement and analysis systems for such networks and describes the architecture of a simulation test-bed for storage of network data for long-term and short-term feature extraction and traffic monitoring which are discussed in the UKLight measurement and monitoring project, MASTS [1]. The Objective of MASTS is to set-up a traffic monitoring system for the UKLIGHT international high capacity experimental network. This facility will allow near real-time view of the various network metrics and enable querying the databases via Web Services. In presence of such high data rates and storage requests, monitoring and measurement becomes a critical yet extremely sophisticated process.

[1] UKLight High-Capacity Network: `www.uklight.ac.uk`

At presence of high data rates on a saturated link, sampling is an important step taken towards reducing the overheads involved in trace collection and traffic analysis for characterisation.

Sampling is the focus of this report. The main disadvantage of sampling is the loss of accuracy in the collected trace when compared to the original traffic stream. In this report some of the techniques of compensation for the loss of details are discussed. To date there has been no work on recovering detailed properties of the original unsampled packet stream, such as the number and lengths of flows. It is important to consider the sampling techniques and their relative accuracy when applied to different traffic patterns. An extension to this work is also discussed, where the applications of network wide core and edge sampling are exploited for network trouble shooting purposes.

Acknowledgements

I would also like to acknowledge continuous support and advice from my supervisor, Dr Miguel Rio and also Professor Saleem Bhatti of St.Andrews University, who have always been there when I was most in need.

I would like to express my gratitude to partners in the MASTS project, especially Dr Andrew Moore of Queen Merry College, who has always been there for advice on different aspects of project work.

I appreciate all the analytical methods and problem solving skills that I learnt whilst visiting Intel Research Cambridge from Dr Gianluca Iannaccone of Intel Research Cambridge and Dr Richard Mortier of Microsoft Research Cambridge. These skills have certainly led me to clarify my objectives and I would like to thank all of those at the Cambridge Computer Laboratory who made my time an amazing one at Cambridge.

I would like to thank Dr Eduarda Mendes Rodrigues of Microsoft Research Cambridge for her support and assistance during the initial stages of my PhD when taking the correct path is critical.

Finally, I wish to acknowledge the personal support from my family and all my friends at the Adaptive Complex Systems Engineering group at UCL, Network Services Research Group and Bloomsbury Fitness.

Publications

- Hamed Haddadi, Lionel Sacks, *Networks Modelling Using Sampled Statistics*, Proceedings of London Communications Symposium: The Annual London Conference on Communication, University College London , 14th-15th September 2006

- Hamed Haddadi, Lionel Sacks, *Passive Monitoring Challenges on High Speed Switched Networks*, Proceedings of London Communications Symposium: The Annual London Conference on Communication, University College London , 8th-9th September 2005

- H Haddadi, E Mendes Rodrigues, L E Sacks, *Development of a Monitoring and Measurement Platform for UKLight High-Capacity Network*, Proceedings of PREP2005 : Postgraduate Research Conference in Electronics, Photonics, Communications and Networks, and Computing Science, University of Lancaster, UK, 30th March to 1st April 2005 (EPSRC Grant Winner)

- H Haddadi, E Mendes Rodrigues, L E Sacks, *Applications of Grid-Probe Technology for Traffic Monitoring on High Capacity Backbone Networks, Data Link Layer Simulation Approach*, Proceedings of IEEE INFOCOM 2005: The Conference on Computer Communications, student workshop, Miami, Florida, USA, March 13th -17th 2005 (Winner of IEEE abstract award and UCL Graduate School Major award)

Contents

List of Figures

Chapter 1

Introduction

1.1 The aims of this report

This report aims to describe the background knowledge and the developed facilities for exploiting traffic flow characteristics within 10 Gbps and similar high capacity switched networks. Using such facility allows validation of statistical compression and feature extraction algorithms which are developed in order to enable monitoring networks over long time scales.

1.2 Motivations of the project

High speed optical switching allows high data transfer rates between several exchange points without the complexity of routing, IP address and port number analysis. The advantages of switched networks have been investigated by researchers who have been looking at Local Area networks (LAN), Metropolitan area Networks (MAN) and Wide Area Networks (WAN).

Ethernet has been one of the most successful standards in the computer networking arena. Today most of data traffic is carried over Ethernet LANs. With the wide adaptation of new Wireless network standards and Ethernet media, extending the range of Ethernet everyday, it is inevitable

that switched networks will play an ever more significant role in the future of telecommunications. Alongside these improvements, the increased need for higher data transfer rates for applications such as streaming audio and video and wide area networks will require thorough research into the characteristics of such networks and their provisioning, utilisation, topology evolution and requirements.

The UK E-Science community, an industrial-academic collaboration initiative, has recently been active in the area of grid networks and high performance computing applications. The purpose of this initiative is to allow researchers throughout UK and Europe to be able to work on multi-site projects generating huge data sets and requiring CPU resources more than those available to a single university or company. Examples of these include the GridPP [2] project at the international particle research laboratory in Geneva, known as CERN, which from year 2007 will begin to produce several Peta bytes of data per year, all of which must be made available to physicists worldwide.

Data volumes like such as these will take a long time to be carried over normal internet links with standard TCP/IP characteristics due to the TCP congestion control algorithms, route failures and packet loss. As a result there is need for design and implementation of more efficient protocols for high bandwidth links. An important proposal regarding this matter was the development of UKLight. UKLight is a national facility in UK to support projects working on developments towards 10 Gbps optical networks and the applications that will use them. UKLight primarily consisted of optical fibre links between University of London Computer Centre, London Point of Presence, NetherLight [3] in Amsterdam and Starlight in Chicago. The links are now extended to more UK universities and UKLight on certain links carries research and production traffic. Figure 1.1 displays the current architecture of the UKLight network.

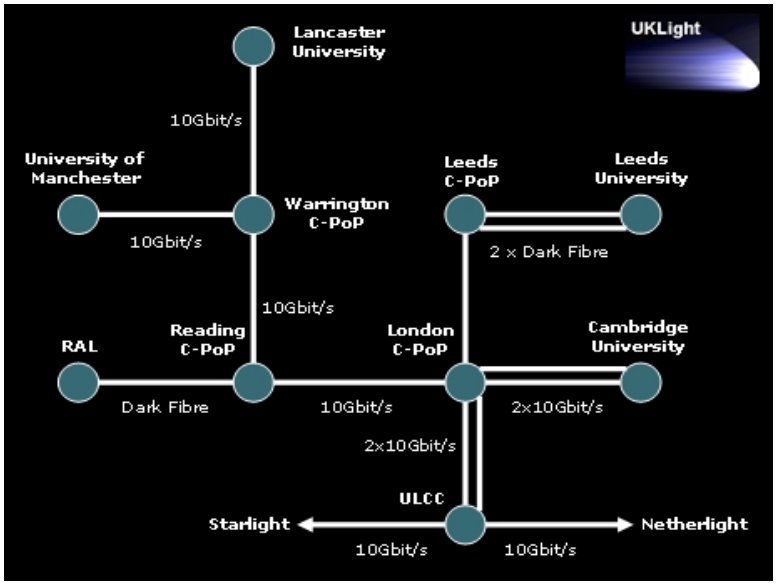

Figure 1.1: UKLight Architecture ([1])

From the operator and users perspective, The monitoring and measurement of network events is an important issue within test networks such as UKLight, where researchers and individuals are allowed to run arbitrary variations of transport, session control and network layer protocols. The UKLIGHT project is envisaged as a platform upon which a rich mix of traffic and data will flow. The users of UKLIGHT are expected to trial many new technologies. For example: ECN, Fast-TCP, Reliable multicast, Diff-Serv, MPLS, and IPv6. Clearly there is a need for tools to evaluate their effectiveness and assessing their impact on the network as a whole. Equally important is the need for work to enable further research and infrastructure offerings which can be enabled through the provision of a system that allows both the interpretation of new service deployments and third-party access to collected data.

UKLight is a switched network which makes it unique in the sense that there is no queueing and routing involved. However, as there is not central point of traffic routing, monitoring the network using router-based informa-

tion, also known as active monitoring, is not possible. This leaves only the option of monitoring the network using the information gathered at the end terminals, which is known as passive monitoring.

The original monitoring architecture proposal for the UKLight is displayed in figure 1.2.

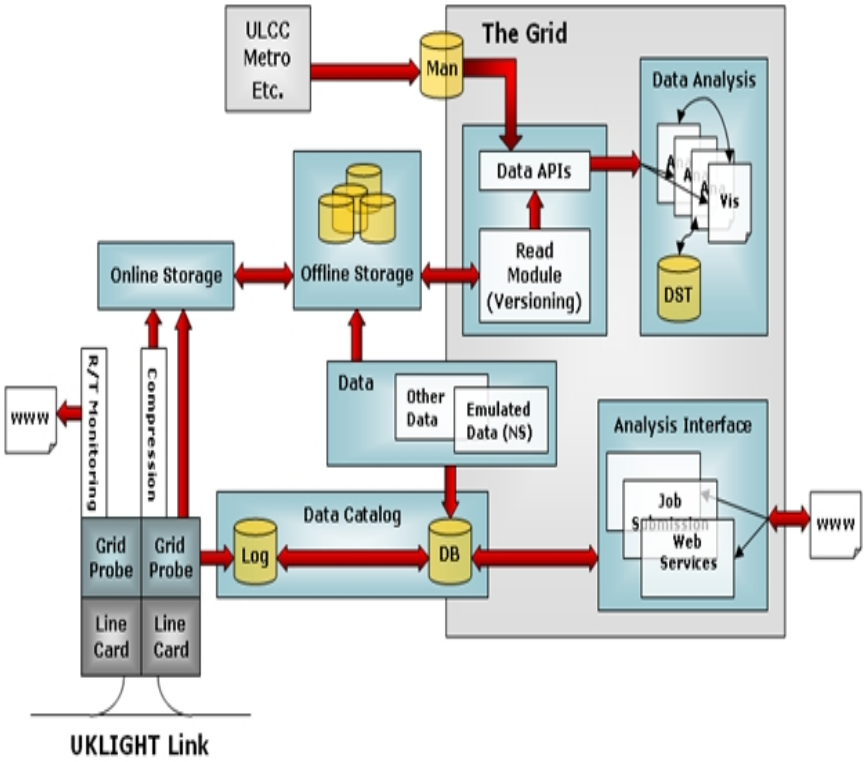

Figure 1.2: UKLight monitoring system architecture ([1])

In the context of passive monitoring, traditionally, flow metrics have been analysed using tools such as NetFlow. NetFlow provides valuable information about network users and applications, peak usage times, and traffic routing [4]. In applications where there is emphasis on packet level data collection, TCP packet trace files are usually collected using variations of libpcap and tcpdump [5]. Tcpdump allows capturing of Ethernet frames at the Network Interface Card (NIC) and stores the data in ASCII format text files. However, still to this date, there is not even a de-facto standard net-

work measurement and monitoring tool deployed world wide, even expensive industrial tools such as HP OpenView [7], only allow for network topology and link monitoring in a more real-time manner, even after adding tools such as Performance Manage Platform (PMP) and do not facilitate analysis of network performance over along time. MRTG is another open-source network monitoring tool. The Multi Router Traffic Grapher (MRTG) is a tool to monitor the traffic load on network links. MRTG generates HTML pages containing PNG images which provide a live visual representation of this traffic [8]. MRTG is only suitable for network bandwidth measurement and it does not allow flow analysis.

There are many packet and traffic capture tools available. However these tools are not capable of handling the high data rates and possibly flow rates on UKLight. Many comparative experiments have been carried on using tcpdump on various systems and on a typical 100Mbps line the capture rates declines rapidly after about 60Mbps send rate [9]. However, the MASTS projects objective is to capture packets at full receive rate and store header data. The captured files are archived to allow for different user-defined queries to be applied on the data. This allows statistical analysis on flow arrival rates, job durations and link transfer times. The objective of this exercise is to let the UKLight user community to be able to view the status of the system over a long period of time and observe how the network topology and usage evolves over time.

These requirements add an extreme edge to the monitoring problem space: high data capture rates, and vast amounts of storage facility. With such requirements, passive monitoring of the network, using collected data at the end nodes is the only feasible way, as adding extra traffic to network for monitoring purposes will make the process more complicated. Collecting the network traffic data will enable the creation of an extensive archive of the traffic flows and their statistics and will bring to the research community an

opportunity for analysis of ethernet and in particular Local Area Network (LAN) traffic's interesting features such as self similarity and long-range dependence of the data. There has been research going on looking into such features in a LAN and it is suggested that LAN traffic demonstrates self-similarity features [10] and this will be looked into in details in the next chapter. LAN traffic consists mostly of human-initiated traffic such as email and web browsing and file sharing. However this is still a grey area in the Grid environment and for a Wide Area Network, such as UKLight which is a switched network extending over thousands of miles, and it will mostly carry machine-generated traffic such as new grid FTP protocols and large database transfers of scientific data.

Another interesting aspect of such networks is the arrival and completion of jobs. There has been research on network traffic and it is suggested that human initiated traffic, such as electronic email and telnet remote login sessions, have Poisson probability distribution function characteristics, but they suggest one should abandon Poisson-based modelling of wide-area traffic for all but user session arrivals [11]. For FTP traffic, it is shown that modelling should concentrate heavily on the extreme upper tail of the largest bursts. A wide-area link such as UKLight might have only one or two such bursts an hour, but they tend to strongly dominate that hours FTP traffic. Vern Paxson and Sally Floyd's research in multiplexed TCP and all-protocol traffic suggests that anyone interested in accurate modelling of wide-area traffic should begin by studying self-similarity [11].

In order to carry out the research into such characteristics on UKLight, there is need for a network simulation platform that closely resembles the topology of UKLight. The objective of this simulation environment is to allow creation of various node topologies, link properties such as delay and bandwidth and various traffic generators such as CBR, Poisson, Exponential and Pareto. This simulation tool has to be able to generate traffic data as

if a network interface card is recording and storing Ethernet frame and IP packets headers.

The concept of measurement is a relatively new concept in networking and it has only been in the last decade that network equipment manufacturers have started taking into account ways of enabling tapping network data. Hence the equipment already deployed in the UKLight network is not able to duplicate the network packets into a packet-sniffing program or to break up the 10G data streams into smaller, more manageable streams. These issues must be taken into consideration in designing the simulation tool and the facility must be there for multiple choices of scenarios. The tool must be able to plot live statistics of the network (such as packet count, flow count, utilisation, delay and etc) and it must also use XML data files to enable the web services to access the simulated traffic archives and retrieval of the data for offline statistical manipulation.

Such a tool will allow a realistic approach to the link monitoring, traffic storage, real time graphical interface, web services queries and archiving issues of the whole project. This is the first step towards the research plans involved in this project, which aims to look into analysis of flow arrival and statistics within a Wide Area Network, which is unique in the sense that it has no routing involved. Hence the links are reserved between individual hosts and there is no routing and congestion involved in the network. Another important characteristic of UKLight is the fact that there is minimal user-initiated traffic, such as email and web browsing, on the network and it will mostly be utilised by machine-to-machine traffic.

There have been attempts to characterise the traffic metrics on networks and to look at the distributions and issues such as self-similarity and Long-Range dependence. However these have mostly been in the context of human-initiated traffic and metrics such as utilisation and delay and there has not been a major attempt to characterise the actual data flows on net-

works, which on a web server hosting frequently accessed web pages can be very different from two machines on a grid network transferring large databases using a proprietary transport protocol such as GridFTP. The research into the above mentioned topics are essential parts of the Measurement at All Scales in Time and Space MASTS project. MASTS is a 1.2 million project that has started in September 2004 and it will record, analyse and publish the traffic and topology evolution throughout the lifetime of UKLight. The project is carried on in collaboration between University College London (EE & CS), LoughboroughUniversity (EE & CS), University of Cambridge (EE & CS) and Endace (Industrial). MASTS is an e-Science of networking project.

MASTS objectives are:

- To measure the development of the topology and traffic in and around the UKLight network
- Develop and Deploy Grid Probe technology on the network
- Develop 'back end' feature extraction and data compression
- Archive flow traces and statistics
- Provide real-time views on specific flows for other UKLight users
- Provide Grid Service based access to archive data

The authors focus within the MASTS project is research into feature extraction algorithms for back end and real time deployment of the system. UKLight

1.3 Layout of the report

This report aims to cover the background on network monitoring, work done so far by the author and the future plans for the project as a whole.

Chapter 2 gives a brief overview of the history and motivations behind network monitoring. It argues why it is becoming more difficult to monitor today and future networks and some challenges that there are on the way.

Chapter 3 discusses the current work in progress on various packet sampling techniques and architectures, from core-based monitoring techniques to edge-based trace collection systems. The chapter goes then onto discussing the advantages and disadvantages of these techniques.

Chapter 4 covers the questions faced by the author when designing a sampling system for different environments, UKLight in particular. In this chapter the methods of generating the original traffic flow statistics from the sampled data statistics and short falls of the current methods are discussed.

Chapter 5 discusses the effects of sampling on a large trace from a major backbone research network and displays the changes in various statistical properties and distribution of variables.

Chapter 6 covers the statistical analysis of the different sampling methods applied and the results of them and discusses the use of adaptive sampling methods.Also discussed is the use of combinational edge-based and core-based sampling for a novel method of generation of Network-wide tomography for troubleshooting and the experiments carried out by the author and plans for future experiments.

In Chapter 7 the conclusion to the report is drawn by discussion of the available tools and the future directions of the project.

Chapter 2

Network Monitoring Principles

This chapter will briefly overview the current trends in research and industry in the field of network monitoring. It includes an argument for the need for network monitoring and a summary of work done by other active researcher in the field and their achievements so far.

2.1 History of computer networks

Prior to the widespread inter-networking that led to the Internet, most communication networks were limited by their nature to only allow communications between the stations on the network. Some networks had gateways or bridges between them, but these bridges were often limited or built specifically for a single use. One prevalent computer networking method was based on the central mainframe method, simply allowing its terminals to be connected via long leased lines. This method was used in the 1950s by Project RAND to support researchers such as Herbert Simon, in Pittsburgh, Pennsylvania, when collaborating across the continent with researchers in Santa Monica, California, on automated theorem proving and artificial in-

telligence [12].

2.1.1 Internet ancestors

At the core of the inter-networking problem lay the issue of connecting separate physical networks to form one logical network. During the 1960s, several groups worked on and implemented packet switching. Donald Davies (NPL), Paul Baran (RAND Corporation) and Leonard Kleinrock (MIT) are credited with the simultaneous invention. The notion that the Internet was developed to survive a nuclear attack has its roots in the early theories developed by RAND. Baran's research had approached packet switching from studies of decentralisation to avoid combat damage compromising the entire network [13].

ARPANET

Promoted to the head of the information processing office at ARPA, Robert Taylor intended to realize Licklider's ideas of an interconnected networking system. Bringing in Larry Roberts from MIT, he initiated a project to build such a network. The first ARPANET link was established between the University of California, Los Angeles and the Stanford Research Institute on 21 November 1969. By 5 December 1969, a 4-node network was connected by adding the University of Utah and the University of California, Santa Barbara. Building on ideas developed in ALOHAnet, the ARPANET started in 1972 and was growing rapidly by 1981. The number of hosts had grown to 213, with a new host being added approximately every twenty days [14].

ARPANET became the technical core of what would become the Internet, and a primary tool in developing the technologies used. ARPANET development was centred around the Request for Comments (RFC) process, still used today for proposing and distributing Internet Protocols and Sys-

tems. RFC 1, entitled "Host Software", was written by Steve Crocker from the University of California, Los Angeles, and published on April 7, 1969. International collaborations on ARPANET were sparse. For various political reasons, European developers were concerned with developing the X.25 networks. Notable exceptions were the Norwegian Seismic Array (NORSAR) in 1972, followed in 1973 by Sweden with satellite links to the Tanum Earth Station and University College London. These early years were documented in the 1972 film Computer Networks: The Heralds of Resource Sharing [14].

X.25 and public access

Following on from DARPA's research, packet switching networks were developed by the International Telecommunication Union (ITU) in the form of X.25 networks. In 1974, X.25 formed the basis for the SERCnet network between British academic and research sites, which would later become JANET. The initial ITU Standard on X.25 was approved in March 1976.

The British Post Office, Western Union International and Tymnet collaborated to create the first international packet switched network, referred to as the International Packet Switched Service (IPSS), in 1978. This network grew from Europe and the US to cover Canada, Hong Kong and Australia by 1981. By the 1990s it provided a worldwide networking infrastructure.

Unlike ARPAnet, X.25 was also commonly available for business use. X.25 would be used for the first dial-in public access networks, such as Compuserve and Tymnet. In 1979, CompuServe became the first service to offer electronic mail capabilities and technical support to personal computer users. The company broke new ground again in 1980 as the first to offer real-time chat with its CB Simulator. There were also the America Online (AOL) and Prodigy dial in networks and many bulletin board system (BBS) networks such as The WELL and FidoNet. FidoNet in particular was popular amongst hobbyist computer users, many of them hackers and

radio amateurs [12].

UUCP

In 1979, two students at Duke University, Tom Truscott and Jim Ellis, came up with the idea of using simple Bourne shell scripts to transfer news and messages on a serial line with nearby University of North Carolina at Chapel Hill. Following public release of the software, the mesh of UUCP hosts forwarding on the Usenet news rapidly expanded. UUCPnet, as it would later be named, also created gateways and links between FidoNet and dial-up BBS hosts. UUCP networks spread quickly due to the lower costs involved, and ability to use existing leased lines, X.25 links or even ARPANET connections. By 1983 the number of UUCP hosts had grown to 550, nearly doubling to 940 in 1984 [12].

2.1.2 Network protocols, TCP/IP

With so many different network methods, something needed to unify them. Robert E. Kahn of DARPA and ARPANET recruited Vint Cerf of Stanford University to work with him on the problem. By 1973, they had soon worked out a fundamental reformulation, where the differences between network protocols were hidden by using a common internetwork protocol, and instead of the network being responsible for reliability, as in the ARPANET, the hosts became responsible. Cerf credits Hubert Zimmerman and Louis Pouzin (designer of the CYCLADES network) with important work on this design [15].

With the role of the network reduced to the bare minimum, it became possible to join almost any networks together, no matter what their characteristics were, thereby solving Kahn's initial problem. DARPA agreed to fund development of prototype software, and after several years of work, the first somewhat crude demonstration of what had by then become TCP/IP

occurred in July 1977. This new method quickly spread across the networks, and on January 1, 1983, TCP/IP protocols became the only approved protocol on the ARPANET, replacing the earlier NCP protocol.

After the ARPANET had been up and running for several years, ARPA looked for another agency to hand off the network to; ARPA's primary business was funding cutting-edge research and development, not running a communications utility. Eventually, in July 1975, the network had been turned over to the Defense Communications Agency, also part of the Department of Defense. In 1984, the U.S. military portion of the ARPANET was broken off as a separate network, the MILNET.

The networks based around the ARPANET were government funded and therefore restricted to noncommercial uses such as research; unrelated commercial use was strictly forbidden. This initially restricted connections to military sites and universities. During the 1980s, the connections expanded to more educational institutions, and even to a growing number of companies such as Digital Equipment Corporation and Hewlett-Packard, which were participating in research projects or providing services to those who were [14].

Another branch of the U.S. government, the National Science Foundation (NSF), became heavily involved in internet research and started development of a successor to ARPANET. In 1984 this resulted in the first Wide Area Network designed specifically to use TCP/IP. This grew into the NSFNet backbone, established in 1986, and intended to connect and provide access to a number of supercomputing centres established by the NSF. It was around the time when ARPANET began to merge with NSFNet, that the term Internet originated,[10] with "an internet" meaning any network using TCP/IP. "The Internet" came to mean a global and large network using TCP/IP, which at the time meant NSFNet and ARPANET. Previously "internet" and "internetwork" had been used interchangeably, and "internet

protocol" had been used to refer to other networking systems such as Xerox Network Services.

As interest in wide spread networking grew and new applications for it arrived, the Internet's technologies spread throughout the rest of the world. TCP/IP's network-agnostic approach meant that it was easy to use any existing network infrastructure, such as the IPSS X.25 network, to carry Internet traffic. In 1984, University College London replaced its transatlantic satellite links with TCP/IP over IPSS.

Many sites unable to link directly to the Internet started to create simple gateways to allow transfer of e-mail, at that time the most important application. Sites which only had intermittent connections used UUCP or FidoNet and relied on the gateways between these networks and the Internet. Some gateway services went beyond simple e-mail peering, such as allowing access to FTP sites via UUCP or e-mail.

The first ARPANet connection outside the US was established to NORSAR in Norway in 1973, just ahead of the connection to Great Britain. These links were all converted to TCP/IP in 1982, at the same time as the rest of the Arpanet [14].

2.2 Watching the network, is there a need?

Traditionally most of the work on research on computer networks has been carried out on subjects such as physical layer, speed improvement, routing protocols and transport protocols. As the number of nodes on the networks, number of different protocols and port numbers used by them, new applications such as multimedia delivery on internet and peer-to-peer networks and the amount of traffic on commercial networks keeps increasing exponentially at a rate much faster than predicted, it is becoming more evident for network administrators and researchers that there are simply not enough tools for measurement and monitoring of traffic and topology on

high speed networks. Even simulation of such networks has been a challenge and to date there is no simulation tool that claims to be able to simulate the internet or even a large Ethernet network precisely. The rapid topology changes and evolution, new applications and trends of use just make it extremely difficult to simulate such networks.

There are basically two types of network monitoring: Active monitoring and passive monitoring. In active monitoring, active traffic such as ping and SNMP data is sent over the network and collected at edge routers. Active monitoring of a network domain introduces some major challenges. Routers of a network domain need to be queried periodically to collect statistics about general status of the system and this huge amount of data has to be stored to obtain useful monitoring information. This increases the overhead for high speed core routers, and restricts the monitoring process from scaling to a large number of flows. To achieve scalability, polling and measurements that involve core routers should be avoided. Hence active network monitoring is limited to periodic route discovery and topology analysis. In passive monitoring the network is analysed only on the edges, without the introduction of an extra traffic, and the network measurement parameters are deduced by applying mathematical formulae on the collected dataset.

The major problem with passive monitoring is storage and processing of many millions of packets which are gathered on edge nodes and the ability of edge "probing points" to capture and store this traffic decays rapidly as the speeds of links go beyond 100s of mega bits per second.

On the other hand, network designers and researchers need to know exactly how the network evolves and behaves over different time scales ranging form few milliseconds for a denial of service attack to few hours or days for a large batch of file transfers between two Grid network nodes. These are just some examples of situations that would require continuous monitoring of the network. As the network's size and usage exceed the average office LAN,

it becomes increasingly difficult to monitor all the activities within the network, with many applications no longer using a common port number (e.g. port 80 for HTTP).

Many network management applications employ measured traffic usage, in packets or bytes, that is differentiated according to header fields into classes at some granularity that depends on the application requirements. Here are some examples:

- "Service development. Service providers track the growth of new applications (as identified by TCP/UDP port numbers) and identify potential new customers that use them (from packet IP addresses not administered in their network).

- Heavy hitters. Determining the dominant components within a class of traffic, for example, the most popular websites, based on the IP destination address of HTTP requests.

- Security applications. Detecting usage indicative of network intrusions, including changes of patterns of usage of specific protocols and TCP/UDP ports, and most active hosts and networks involved.

- Network engineering. A service provider determines the intensities of traffic between sets of source and destination addresses that is carried over a congested link. This information could be used to examine the feasibility of rerouting portions of the traffic away from the congested link.

- Chargeback. A corporate intranet apportions its costs to constituent organisations, based on usage that originates in the organisations IP address ranges.

- Customer billing. A service provider charges customers, as identified by IP address, for byte usage. The rate of charge may depend on ap-

plication type (as identified by TCP/UDP port numbers). Charging based on remote address (e.g., whether on or off the providers network) also has been proposed. Information on some commercial examples of the use of flow records for billing purposes can be found in [30]. The use of packet samples for billing is proposed for InMons sFlow [31]. The accuracy requirements of these applications are quite different; the list above is (roughly) ordered by stringency, where customer billing is the most stringent. There are strong legal reasons for not overbilling customers and it would not be good customer relations. (Some regulatory environments may prohibit billing on estimated usage.) On the other hand, network management applications can probably tolerate errors of quite a few percent in estimating usage by a class of traffic. The ramifications of sampling for the estimation of network usage are a central theme of this review.

- Path measurement. Duffield et al. [32] describes a method to measure entire paths of packets through the network. As well as determining per class usage along paths, this method enables the passive measurement of network path performance, route troubleshooting and network attack tracing.

- Traffic structure. Sampling can present a challenge to the determination of detailed structural properties of traffic (such as the duration of traffic flows) or the composition of application transactions (such as a Web browsing session) in terms of multiple application flows.An inference method whereby sampled measurement are used to infer detailed structural properties of the original unsampled flows of traffic is described in [32]."

2.3 Current research in measurement and monitoring

Network researchers have adopted two distinct approaches to data collection. The first approach uses an *active* measurement system to inject probe traffic into the network and then extrapolate the performance of the network from the performance of the injected traffic. The second approach is that of *passively* observing and recording network traffic. These passive measurement systems use the recorded traffic to characterise both the applications and the networks performance. They record and archive full traces, which in turn can be later used for re-analysis. One drawback is that they generate a large amount of measurement data. Due to the quantity of data produced, recording traces from very high bandwidth links is a serious challenge. As a result, global observations have often been addressed by inference techniques, and not by exhaustive passive monitoring of every link in a network.

OC3MON is a well-known passive monitoring system for OC-3 links (155 Mbps) described in [24]. It collects packetlevel traces or flow-level statistics. Packet-level traces can be collected only for a limited amount of time (only a few minutes at a time), while flow-level statistics can be collected on a continuous basis. It has been deployed at two locations in the MCI backbone network to investigate daily and weekly variations in traffic volume, packet size distribution, and traffic composition in terms of protocols and applications. OC3MON has now been extended to support OC-12 and OC-48 links.

Passive monitoring systems require specific hardware to collect data on the network. In the case of OC3MON, data capture relies on tapping the fibre through a dedicated network interface card. There are several projects which combine both active and passive measurement. The

NetScope project [25] collects measurements from the AT&T network in order to study the effects of changing network routes and router configuration. Using NetFlow measurements from routers, the traffic demand for the entire network is derived. The traffic demand is used in simulation to determine the effects of changing the network configuration. As part of an ongoing effort to develop better network measurement tools, a passive monitoring system called PacketScope has been developed and used to collect and filter packet-level information. The NAI (Network Analysis Infrastructure) project measures the performance of the VBNS and Abilene networks. This system collects packet traces, active measurements of roundtrip delay and loss, and BGP routing information. All of the 90-second-long packet traces from this project are available on their web site. Some routers have built-in monitoring capabilities.

Cisco routers have NetFlow [17]. It collects information about every TCP and UDP flow on a link. Juniper routers have a set of accounting tools to collect similar statistics as NetFlow. There are other stand-alone commercial products for passive monitoring, such as Niksuns NetDetector and NetScouts ATM Probes. These systems, however, are limited to OC-3 or lower link speeds, and are thus not adequate for Internet backbone links.

The Sprint [29] monitoring infrastructure, called IPMON, is similar to the OC3MON system, but with extended capabilities that allow it to collect packet traces at up to OC-48 link speeds (2.48 Gbps) for a period of at least several hours. The range of observable metrics is wider than with the above systems thanks to timestamps synchronised to within 5 νs of a global clock signal. Sprint researchers have deployed our monitoring infrastructure on multiple OC-3, OC-12, and OC-48 bidirectional links in 4 POPs in the Sprint IP backbone network, and collected weeks of traces. However this work had been done in an environment where Cisco NetFlow tools have been present in order to enable a comparison of results with those recorded at

the routers. Another limitation is the bandwidth increase on the UKLight scenario which goes to 10 Gbps links which use OC-192 cards.

Flow characterisation and packet classification is an area which is closely related to any monitoring activity. Network operators, and exceedingly users of services, would like to know how their network or connection to provider is utilised, what applications are consuming the bandwidth and what ports they operate on. This is a major area of research and it leads to activities such as billing, network tomography and anomaly detection.

A better understanding of the nature and origin of flow rates in the Internet is important for several reasons. First,in order to understand the extent to which application performance would be improved by increased transmission rates, one must first know what is limiting their transmission rate. Flows limited by network congestion are in need of drastically different attention than flows limited by host buffer sizes. Further, many router algorithms to control per-flow bandwidth algorithms have been proposed, and the performance and scalability of some of these algorithm depends on the nature of the flow rates seen at routers. Thus, knowing more about these rates may inform the design of such algorithms. Finally, knowledge about the rates and their causes may lead to better models of Internet traffic. Such models could be useful in generating simulation workloads and studying a variety of network problems and are studied in [35]. The authors claim their findings confirm what has been observed previously, the distribution of flow rates is skewed, but not as highly skewed as flow sizes and that flow rates strongly correlated with flow sizes.

This is strong evidence that user behaviour, as evidenced by the amount of data they transfer, is not intrinsically determined, but rather, is a function of the speed at which files can be downloaded. UKLight carries traffic on 10 Gbps links and the monitoring of these links is the main objective of the project. However the process of capture and store of frame headers

even at 1Gbps speed is cumbersome. Researchers at Intel laboratories have been working on a continuous Monitoring project called CoMo [36]. CoMo has been designed to be the basic building block for a network monitoring infrastructure that will allow researchers and network operators to easily process and share network traffic statistics over multiple sites.

The architecture of CoMo, as shown in 2.1, is designed to compute and report various performance metrics while sustaining high speed traffic collection. CoMo also provides a query interface to allow users to elicit the system to export the results of the measurement performed. The suitability of CoMo for this project is currently being investigated by the author and researchers at Loughborough University. The capture rates achieved are up to around 700 Mbps.

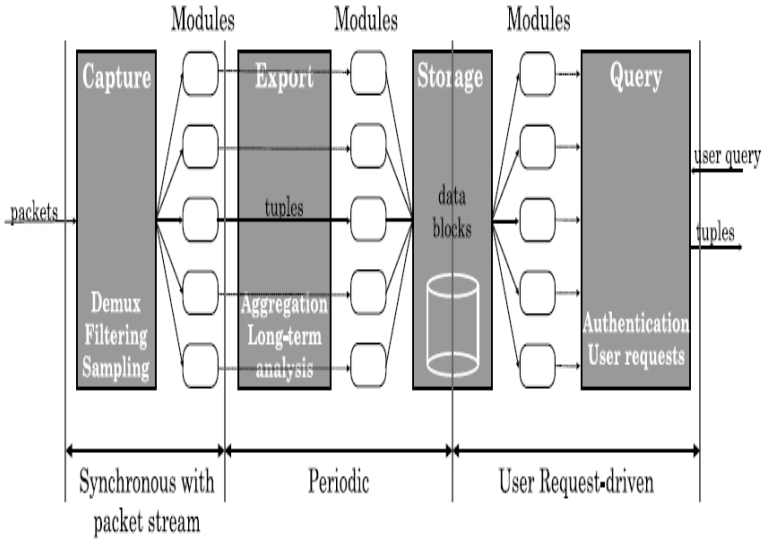

Figure 2.1: CoMo Architecture (Figure courtesy of Intel Research [36]

2.4 Tools and techniques of network measurement

In classic network measurement applications such as SNMP, traceroute and snort, attention was paid mostly on factors such as round-trip delay, loss

and throughput. This trend has mainly been caused by a sudden increase in the number of users of internet and the exponential growth of the volume of traffic carried across networks. Many tools use SNMP, RMON [22], or NetFlow [17], which are built-in functionality for most routers. Using these mechanisms, a centralised or decentralised model can be built to monitor a network. The centralised approach to monitor network latency, jitter, loss, throughput, or other QoS parameters suffers from scalability.

On the other hand with the edge-based passive monitoring there is need to store, archive and analyse extremely large data sets. These data sets are sometimes collected over a long time duration in order to look at factors such as burstiness, self-similarity and long-range dependence of network traffic. Researchers have been working on the self-similar nature of the internet traffic using sampled data and it has been proven that internet traffic is self-similar at different times and scales [10]. A detailed list of network monitoring tools can be found in [23]. In this section the most relevant ones are discussed. The following description are all extracted from external sources, typically the standards, and referenced respectively.

2.4.1 SNMP

The simple network management protocol (SNMP) forms part of the internet protocol suite as defined by the Internet Engineering Task Force. The protocol is used by network management systems for monitoring network-attached devices for conditions that warrant administrative attention.

Management Information Base (MIBs)

The SNMP protocol's extensible design is achieved with management information bases (MIBs), which specify the management data of a device subsystem, using a hierarchical namespace containing object identifiers, implemented via ASN.1. The MIB hierarchy can be depicted as a tree with

a nameless root, the levels of which are assigned by different organisations. This model permits management across all layers of the OSI reference model, extending into applications such as databases, email, and the Java EE reference model, as MIBs can be defined for all such area-specific information and operations.

Architecture

SNMP framework consists of master agents, subagents and management stations. A master agent is a piece of software running on an SNMP-capable network component (say, a router) that responds to SNMP requests made by a management station. Thus it acts as a server in client-server architecture terminology or as a daemon in operating system terminology. A master agent relies on subagents to provide information about the management of specific functionality. Master agents can also be referred to as Managed objects. A subagent is a piece of software running on an SNMP-capable network component that implements the information and management functionality defined by a specific MIB of a specific subsystem (e.g., the ethernet link layer). Some capabilities of the subagent are gathering information from managed objects, configuring parameters of the managed objects, responding to managers' requests, and generating alarms (or traps). The manager or management station is the final component in the SNMP architecture. It functions as the equivalent of a client in the client-server architecture. It issues requests for management operations on behalf of an administrator or application, and receives traps from agents as well.

The SNMP protocol

The SNMP protocol operates at the application layer (layer 7) of the OSI model. It specified (in version 1) five core protocol data units (PDUs):

1. GET REQUEST, used to retrieve a piece of management information.

2. GETNEXT REQUEST, used iteratively to retrieve sequences of management information.

3. GET RESPONSE

4. SET, used to make a change to a managed subsystem.

5. TRAP, used to report an alert or other asynchronous event about a managed subsystem.

In SNMPv1, asynchronous event reports are called traps while they are called notifications in later versions of SNMP. In SMIv1 MIB modules, traps are defined using the TRAP-TYPE macro; in SMIv2 MIB modules, traps are defined using the NOTIFICATION-TYPE macro. Other PDUs were added in later versions, including:

1. GETBULK REQUEST, a faster iterator used to retrieve sequences of management information.

2. INFORM, an acknowledged trap.

The first RFCs for SNMP, now known as Simple Network Management Protocol version 1, appeared in 1988:

- RFC 1065 Structure and identification of management information for TCP/IP-based internets

- RFC 1066 Management information base for network management of TCP/IP-based internets

- RFC 1067 A simple network management protocol

Version 1 has been criticised for its poor security. Authentication of clients is performed only by a "community string", in effect a type of password, which is transmitted in cleartext. The '80s design of SNMP V1 was done by a group of collaborators who viewed the officially sponsored /OSI/IETF/NSF

(National Science Foundation) effort (HEMS/CMIS/CMIP) as both unimplementable in the computing platforms of the time as well as potentially unworkable. SNMP was approved based on a belief that it was an interim protocol needed for taking steps towards large scale deployment of the Internet and its commercialisation. In that time period Internet standard authentication/security was both a dream and discouraged by focused protocol design groups. The Internet Engineering Task Force (IETF) recognises Simple Network Management Protocol version 3 as defined by RFC 3411RFC 3418 (also known as STD0062) as the current standard version of SNMP as of 2004. The IETF considers earlier versions as "Obsolete" or "Historical" [16].

Usage examples

snmpwalk

The output below show an example of an snmpwalk (snmpwalk is a Net-SNMP application) performed on a router, and shows general information about the device.

```
snmpwalk -c public punch system
SNMPv2-MIB::sysDescr.0 = STRING:unis see Cisco Internetwork Operating System Software
IOS (tm) C2600 Software (C2600-IO3-M), Version 12.2(15)T5, RELEASE SOFTWARE (fc1)
TAC Support:  http://www.cisco.com/tac
Copyright (c) 1986-2003 by cisco Systems, Inc.
Compiled Thu 12-Jun-03 15:49 by eaarm
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.9.1.187
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks:  (835747999) 96 days, 17:31:19.99
SNMPv2-MIB::sysContact.0 = STRING: wikiuser
SNMPv2-MIB::sysName.0 = STRING: punch
SNMPv2-MIB::sysLocation.0 = STRING: test
SNMPv2-MIB::sysServices.0 = INTEGER: 78
```

```
SNMPv2-MIB::sysORLastChange.0 = Timeticks:  (0) 0:00:00.00
```

Router graphing software

A lot of data about the performance, load and error rates of network elements like routers and switches can be gathered through SNMP. There are a number of tools which gather this data on a regular basis and which can produce various kinds of graphs from it. Such graphs can be interpreted by network adminstrators to evaluate a network's performance, identify potential bottlenecks and help in redesigning a network. Example tools of this type are MRTG and Cacti [16].

Proxy agent Normally, a network management system is able to manage device with SNMP agent installed. However in the absence of the SNMP agent, it can be managed with the help of a proxy agent. The SNMP agent associated with the proxy policy is called a proxy agent, or commercially a proxy server. The proxy agent monitor non-SNMP Community with non-SNMP agents and then converts the objects and data to SNMP compatible objects and data to be fed to an SNMP manager [16].

2.4.2 CISCO NetFlow

As IP traffic continues its explosive growth across today's networks, enterprise and service providers must be able to characterise this traffic and account for how and where it flows. The challenge, however, is finding a scalable, manageable, and reliable solution to provide the necessary data to support these opportunities. Cisco IOS NetFlow technology is an integral part of Cisco IOS Software that collects and measures data as it enters specific routers or switch interfaces.

By analysing NetFlow data, a network engineer can identify the cause of congestion; determine the class of service (CoS) for each user and application; and identify the source and destination network for traffic. NetFlow

allows extremely granular and accurate traffic measurements and high-level aggregated traffic collection. Because it is part of Cisco IOS Software, NetFlow enables Cisco product-based networks to perform IP traffic flow analysis without purchasing external probes–making traffic analysis economical on large IP networks [16].

Usage scenarios

Network application and user monitoring: NetFlow data enables users to view detailed, time- and application-based usage of a network. This information allows planning and allocation of network and application resources, including extensive near real-time network monitoring capabilities. It can be used to display traffic patterns and application-based views. NetFlow provides proactive problem detection, efficient troubleshooting, and rapid problem resolution. This information is used to efficiently allocate network resources and to detect and resolve potential security and policy violations.

Network planning: NetFlow can be used to capture data over a long period of time, which enables users to track and anticipate network growth and plan upgrades to increase the number of routing devices, ports, or higher-bandwidth interfaces. NetFlow services data optimises network planning, which includes peering, backbone upgrade planning, and routing policy planning. It minimises the total cost of network operations while maximising network performance, capacity, and reliability. NetFlow detects unwanted WAN traffic, validates bandwidth and Quality of Service (QoS), and enables the analysis of new network applications. NetFlow will offer valuable information to reduce the cost of operating the network.

Security analysis: NetFlow data identifies and classifies Denial of Service (DoS) attacks, viruses, and worms in real-time. Changes in network behaviour indicate anomalies that are clearly demonstrated in NetFlow data. The data is also a valuable forensic tool to understand and replay the history

of security incidents.

IP accounting and Usage-based billing: NetFlow technology also enables customers to implement usage-based billing, providing them with the ability to implement competitive pricing schemes and premium services. In addition to measurement and billing, NetFlow also performs strategic analysis on their point-of-presence (POP) traffic for network planning, acceptable usage policy enforcement, or service-level management (SLM). Customers can, therefore, use NetFlow to track IP traffic flowing into or out of their server farms for capacity planning or to implement usage-based billing.

Traffic engineering: NetFlow can measure the amount of traffic crossing peering or transit points to determine if a peering arrangement with other service providers is fair and equitable.

NetFlow operation and architecture

NetFlow includes three key components that perform the following capabilities:

- Flow caching analyses and collects IP data flows entering router or switch interfaces and prepares data for export. It enables the accumulation of data on flows with unique characteristics, such as IP addresses, application, and CoS. Flexible flow data is now available using the latest NetFlow v.9 export data format. NetFlow supports key technologies, including IPv4, IPv6, Multicast, and Multiprotocol Label Switching (MPLS).

- FlowCollector and Data Analysis captures exported data from multiple routers and filters and aggregates the data according to customer policies, and then stores this summarised or aggregated data. Users can leverage Cisco NetFlow collector as a flow collector, or they can opt for a variety of third-party partner products. A Graphical user

interface displays and analyses NetFlow data collected from FlowCollector files. This allows users to complete near-real-time visualisation or trending analysis of recorded and aggregated flow data. Users can specify the router and aggregation scheme and desired time interval.

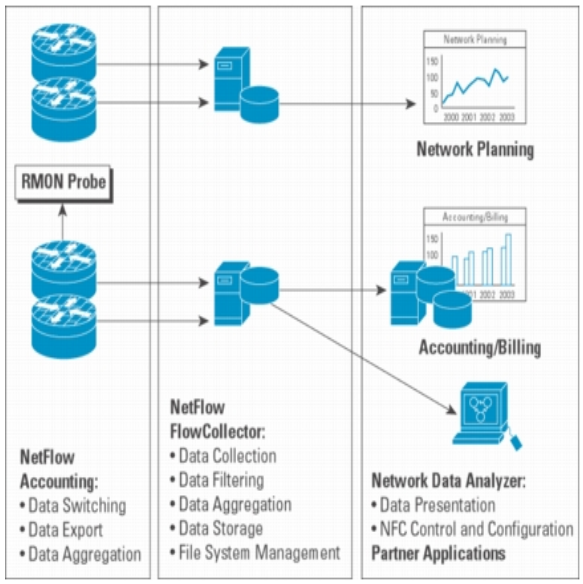

Figure 2.2: Cisco IOS NetFlow Infrastructure([17])

Typical flow analysis information found in a NetFlow data record includes [17]:

- "Source and destination IP address
- Source and destination TCP/User Datagram Protocol (UDP) ports
- Type of service (ToS)
- Packet and byte counts
- Start and end timestamps
- Input and output interface numbers
- TCP flags and encapsulated protocol (TCP/UDP)

- Routing information (next-hop address, source autonomous system (AS) number, destination AS number, source prefix mask, destination prefix mask)"

2.5 Summary

Network monitoring is an essential part of the system life cycle of a network of any size. There are various tools available which produce course-grained aggregate statistics of the network status. Even though these tools are available and deployed, access to detailed statistics of signatures of various networks events such as intrusions, worms and denial of service attacks. Access to such statistics is only possible by use of rigorous mathematical operations on gathered traces of network packets at an aggregation point and a routers. An important step in this path is sampling. Sampling is used for selecting a small portion of the traffic which will enable us to analyse the characteristics of the network. Sampling is discussed in the next chapter.

Chapter 3

Measurement and Sampling

The aim of network measurement is to provide the data for network control, enabling the service provider to characterise the state of the network, the demands of traffic and its consumption of network resources, and the performance experienced by traffic on the network. Measurement systems monitor the system response to reconfiguration to determine if corrective actions are required. Actions operate over a range of time scales, from the deployment of new network infrastructure over a period of months, through tracing network attacks over minutes or hours, to control of traffic flows in close to real time, for example.

3.1 Data analysis dilemma

Internet service providers have in one sense too little data at their disposal, while in another sense they have too much. Too little, because it is not always possible to directly measure quantities of interest, these being only components in aggregate measurements. For example, troubleshooting packet loss requires knowing network performance on individual links, while in practice it may only be feasible to measure performance between two hosts, that is, the composite performance along a path that comprises sev-

eral links. A recent response to the problem of too little data was to develop tomographic methods for inferring individual components from collections of aggregate measurements [32].

When troubleshooting loss, correlating performance measures along intersecting network paths reveals the performance on the intersection of those paths. The too much data problem is that the volumes collected are truly enormous. A large service provider may collect data from tens of thousands of network interfaces. A single high speed network interface could in principle generate hundreds of gigabytes of (unsampled) flow statistics per day if fully utilised, while the whole network might generate several gigabytes of simple network management protocol (SNMP) statistics per day. Furthermore, the rate of data collection is growing, due to the requirement for ubiquitous fine grained measurements. As a result, routers and switches are being equipped with increasingly sophisticated measurement capabilities, providing ever more data to the service providers.

3.2 Data reduction by sampling

The main resource constraint for the formation of flow statistics is at the router flow cache. To perform lookup of packet keys and counter increment at line rate would require the flow statistics to be stored in fast memory. However, core routers will carry increasingly large number of concurrent flows, necessitating large amount of fast memory: this would be expensive. By sampling the packet stream in advance of the construction of flow statistics, the time window available for flow cache lookup is prolonged, enabling storage to be carried out in slower, less expensive, memory.

For many applications, measured data must be transmitted to collection points for storage and analysis. The massive volume of data has cost ramifications for the collection infrastructure. First, processing and storage resources on the routers and switches are comparatively expensive and

scarce in practice; they are already employed in the regular work of routing and switching packets. Second, the transmission of measured data to the collection points can consume significant amounts of network bandwidth. Third, sophisticated and costly computing systems are required for analysis and storage of the data. These three factors motivate data reduction. However, there is an inherent tension between reducing data, on the one hand, and supplying sufficiently detailed measurements for applications, on the other. This tension is most evident at the observation point, where resources are typically the least available. Data reduction is preferably carried out online in a single pass through the traffic stream to avoid buffering and reprocessing. Three methods are commonly employed:

- Aggregation: The combination of several data into a single composite, the components of which are then discarded. Aggregation is commonly additive, for example, finding the total traffic from a set of sources or over a time interval. Aggregates are used to provide a compact data summary when it is acceptable to lose visibility of the aggregates components.

- Filtering: Selection of data based on the data values; unselected data are discarded. For example, traffic from a given source is selected. Filtering is useful to drill down to a subset of traffic of interest, once that subset has been identified.

- Sampling: Random or pseudorandom selection of data; unselected data are discarded. For example, simple random sampling of packets. There is overlap between filtering and sampling: implementations of sampling may be filters by the above definitions, albeit with an exceedingly complex selection rule.

The important factor that distinguishes these three methods is that filtering and aggregation require knowledge of the traffic's features of interest

in advance, whereas only sampling allows the retention of arbitrary detail while at the same time reducing data volumes. [32]

3.3 Packet monitoring

Packet monitoring entails passively copying a stream of packets, then selecting, storing, analysing and/or exporting information on these packets. Until recently packet monitoring was performed exclusively by special purpose hosts installed in the network; A copy of the packet stream is brought to a monitor in one of three ways: by copying the physical signal that carries the packets (e.g., with an optical splitter) and bringing the signal to an interface on the monitor; by attaching the monitor to a shared medium that carries the traffic; by having a router or switch copy packets to an interface to which the monitor is attached. Packet monitors have to cope with some formidable demands on their resources, particularly on the processing bandwidth needed to work at the full line rate of increasingly high speed links.

Restricting data capture to some initial number of bytes of the packet is a common way to control data bandwidth at the monitor. This is a reasonable solution, since the IP header and other protocol header information is located at or near the start of the packet. Even so, widespread continuous collection, transmission and storage of unreduced packets has been infeasible for a number of years due to the immense volumes of data relative to the capacity of systems to collect them; see [18] for an early reference and [19] for a recent one.

Collection of full packet header traces is feasible only for limited durations. Instead, for applications that require continuous monitoring over an extended period, it is common to perform analysis at or near the monitor by forming flow records or other aggregate statistics, or a more general stream querying functionality (see [36]). Collection of packet IP and transport head-

ers is commonly performed using tcpdump [5] or its variant windump [6]. Depending on the traffic load and processing power at the measurement host, these tools may also be able to capture parts of the packet payload.

In network elements, deployment of packet monitors is limited by equipment availability and administrative costs. A more recent approach to packet monitoring was to embed the passive measurement functionality within network elements such as routers and switches. Once packet monitoring capabilities become available in network elements, packet measurement can become ubiquitous in the network. However, little or no capabilities for measurement analysis are expected to be available in routers and switches, because they generally lack the additional computational resources for this purpose. Instead, some form of data reduction is required, both in the selection of information from packets and in the selection of packets to be reported on.

Some packet sampling capabilities are becoming available in routers, for example, sampling in InMons sFlow [31]. Packet selection capabilities for network elements are currently being standardised by the Packet Sampling (PSAMP) Working Group of the Internet Engineering Task Force (IETF). The aim of this work is to define a set of packet selection capabilities which are simple enough to be ubiquitously deployed, yet rich enough to support the needs of measurementbased network management applications. Although specific selection operations are yet to be finalised, it is likely that this will include filtering and various forms of sampling.

3.4 Flow records

A flow of traffic is a set of packets with a common property, known as the flow key, observed within a period of time. Many routers construct and export summary statistics on packet flows that pass through them. Ideally, a flow record can be thought of as summarising a set of packets that arises

in the network through some higher level transaction, for example, a remote terminal session or a web page download. In practice, the set of packets that are included in a flow depends on the algorithm used by the router to assign packets to flows. The flow key is usually specified by fields from the packet header, such as the IP source and destination address and TCP/UDP port numbers. Flows in which the key is specified by individual values of these fields are often called raw flows, as opposed to aggregate flows in which the key is specified by a range of these quantities.

Flow statistics are created as follows: When a packet arrives at the router, the router determines if the flow is active, that is, if statistics are currently being collected for the packets key. If not, it instantiates a new set of statistics for the key. The statistics include counters for packets and bytes that are updated according to each packet matching the key. When the router judges that the flow is terminated, the flows statistics are exported in a flow record and the associated memory is released for use by new flows [32].

A router at the core of an internet link is carrying a large number of flows at any given time. this pressure on the router entails the use of strict rules in order to export the statistics and keep the router memory buffer and CPU resources available to deal with changes in traffic patterns by avoiding the handling of large tables of flow records. Rules for expiring NetFlow cache entries include:

- Flows which have been idle for a specified time are expired and removed from the cache (15 seconds is default)
- Long lived flows are expired and removed from the cache (30 minutes is default)
- As the cache becomes full a number of heuristics are applied to aggressively age groups of flows simultaneously
- TCP connections which have reached the end of byte stream (FIN) or

which have been reset (RST) will be expired

Flow definition schemes have been developed in research environments and are being standardised by the IP Flow Information Export (IPFIX) Working Group of the IETF. Examples of flow definitions employed as part of network management and accounting systems can be found in Cisco's NetFlow [17].

A flow record typically includes the properties that make up a flow's defining key, the arrival times of the first and last packets, and the number of packets and bytes in the flow. Flow records yield considerable compression of information, since a flow is summarised in a fixed length record, regardless of the number of packets in the flow. The trade-off is loss of detail of the timing of packets within the flow. The compression factor depends on the composition of traffic: it is greater for long flows and smaller for short flows. For traffic mixes observed in backbone traffic, byte compression factors for IP and transport headers versus NetFlow records of 25 or more are commonly attainable [32].

3.5 Uniform sampling techniques

This section reviews classical sampling methods [systematic, simple random and stratified sampling and their common applications in passive Internet measurement.

3.5.1 Systematic sampling

In count-based systematic sampling, the triggers are $i_n = nN + i_0$, the occurrence of objects with integer period $N > 0$. In the simplest case, the object in is selected. More generally, $M \leq N$ objects $i_n, ..., i_{n+M1}$ are selected. An example is the capture of subsets of successive packets. Sampling of consecutive packets may be useful for understanding the detailed

dynamical behaviour of packet streams. In time-based systematic sampling, triggers fire at times $\tau_n = nT + \tau_0$. Selection takes place after each trigger has fired, for example, selection of the next arriving object or all objects arriving within a time t of the trigger. Systematic sampling is very straight forward to implement: Set a counter to the sampling period, decrement on each packet, select a packet on reaching zero, then reset the counter and repeat.

However, systematic sampling is vulnerable to bias if the objects being sampled exhibit a period which is rationally related to the sampling period, since samples are taken only at a discrete set of phases within the period. Potential sources of periodicity are timers in protocols and periodically scheduled applications. A further drawback is that periodic sampling is to some extent predictable and, hence, open to deliberate manipulation or evasion.

3.5.2 Random additive and simple random sampling

The potential problems of systematic sampling are avoided by suitable use of random additive sampling. Here, the intervals between successive triggers are independent random variables with a common distribution. (Periodic sampling is a degenerate case where the random variable takes a constant value.) The advantages of random additive sampling for Internet measurement were highlighted by Paxson [35]. It avoids synchronisation problems. Choosing the intervals to be geometrically distributed (for countbased sampling) or exponentially distributed (for timebased sampling) avoids predictability. "Furthermore, Wolffs Poisson arrivals see time averages (PASTA) property ensures that any empirical mean over sampled objects is an unbiased estimator of the corresponding population mean" [21].

For these reasons, random additive sampling is recommended in standards for performance metrics. A simple implementation of random additive

sampling is to generate, immediately following a given trigger, the length of the interval until the next trigger. However some generated intervals will not fit in storage unless a cutoff is applied. "The special case of geometric random additive sampling with mean intersample count m is equivalent to simple random sampling with probability $1/m$. It can be implemented by making a sampling decision for each object, although this is computationally more costly than generating random intersample times" [32].

3.6 Summary

This chapter has provided the knowledge to standard measurement and monitoring techniques available to network researchers and operators. The need for detailed statistics of network operations has recently increased following the growth of applications on internet ranging from Voice over IP, streaming media and content distribution to peer-to-peer applications. Operators have increased the accuracy of their traffic measurement techniques in order to be able to classify services and charge for the high-priority and QoS services. The provision of such billing systems requires adequately detailed reports on bandwidth usage and traffic generation by various users for the operator. It is becoming increasingly important to be able to infer the characteristics of the traffic over the network over different time scales to be able to detect any anomalies resulting form various attacks and usage increases. In the next chapters some of these inference techniques are looked into more carefully and novel inference and trouble shooting methods are discussed.

Chapter 4

Measurements on Large Networks

Sampling is a method of reducing the amount of data collected and reported at the measurement points on networks. However the statistics gathered from these datasets are based on a sampled subset of the real data stream. Consider a constant sampling of 1 in 1000 packets which is currently in use in Cisco Netflow routers; in this case, if a flow is shorted than 1000 packets, there is a probability that it never gets sampled. Many worms, ICMP packets and HTTP requests and replies are shorted than 1000 packets. So clearly, by simply multiplying the final statics by the inverse of the sampling rate (1000 in this case), the network manager will not get the individual flow properties, though it is possible to get a fairly accurate measure of the general traffic characteristics like average throughput, total number of active flows and the distribution of their sizes. In this section, the impact of sampling and inversion by multiplication on network traffic is discussed.

4.1 Analysis of flows on GEANT

Network simulation requires a thorough understanding of the characteristic of the flows on a network, the activity levels of hosts behind networks and the amount of data and the frequency of its tansmission from sources behind a network to the destinations outside, possibly worldwide. In order to get a feel for the nature of traffic on a network such as GEANT, the Netflow data was collected and analysed for the 1 day period of 24^{th} November 2004. The results are those of sampled NetFlow records, an example of which is displayed below, both lines span over two lines.

==========

Timestamp Duration Proto Source IP/Mask Port Destination IP/Mask Port Bytes Packets SrcAS DstAS TCP Flags Exporter Addr Next Hop Addr Engine Type/ID Input/Output Iface Index

Sat Jan 1 00:02:05 2005 1104537725.850 0.000 6 193.170.0.0/15 38877 194.85.32.0/20 80 52 1 1853 2603 16(—-A—) 127.0.0.1 62.40.96.56 0/0 61/87

==========

These results are divided into two sections and discussed in this section. Figure 4.1 displays the topology of GEANT network as of April 2004. GEANT is a pan-European multi-gigabit data communications network, reserved specifically for research and education use. It is creating the biggest interconnected community of scientists and academics in the world today, enabling them to share and distribute research data faster than ever before. It delivers exciting benefits to its users and will play an important role in shaping the future of European science.

GEANT is the latest generation of pan-European research network infrastructure and is one of the most advanced and reliable networks in the world. It provides the highest capacity, and offers the greatest geographic coverage, of any network of its kind in the world. By the end of the project in June 2005, GEANT served over 3,500 research and education institutions

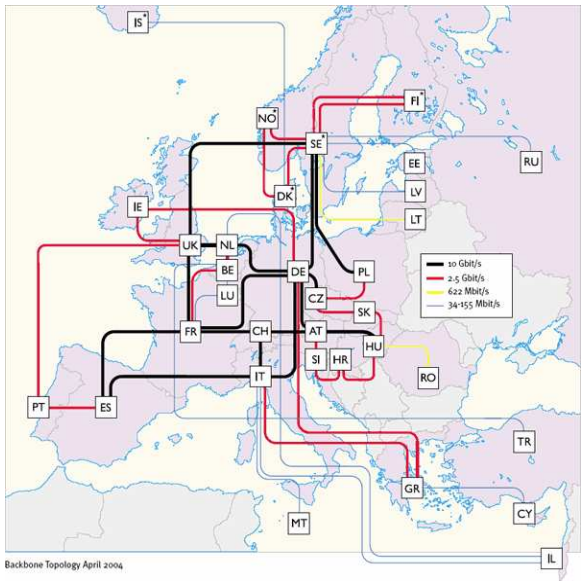

Figure 4.1: GEANT network topology([37])

in 34 countries through 30 national and regional research and education networks [37].

4.1.1 Node activity summaries

The following figures display the activity of the nodes behind 2 of the 23 GEANT routers for comparison purposes. Figure 4.2 displays the number of flows that have been seen from the hosts, *x-axis sorted by the first 3 digits of the IP address , from 0 to 255, covering the whole IP address spectrum, of the IP prefix.* behind the Slovakia router.

Slovakia, being one of the smaller contributers to the traffic on GEANT, has a more limited number of hosts on a small range of IP prefixes. It can be observed that when a simulation scenario is to be created, the distribution of source addresses, and how they differ for different locations on network plays a significant role. Figure 4.3 displays the number of connections that have been made by the hosts behind the UK router [x-axis sorted by the first 3 digits of the IP address , from 0 to 255, covering the whole IP address

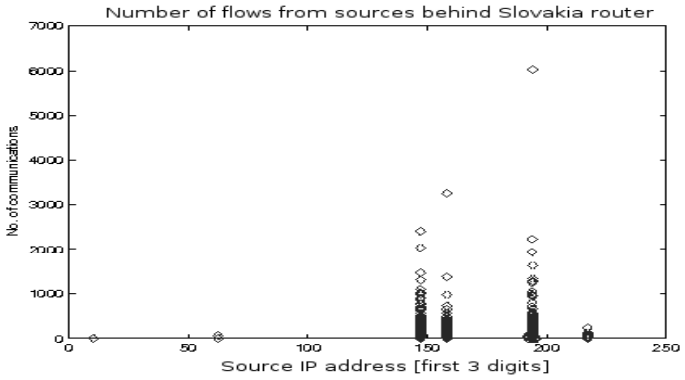

Figure 4.2: Number of flows from source hosts behind router SK1

spectrum, of the IP prefix].

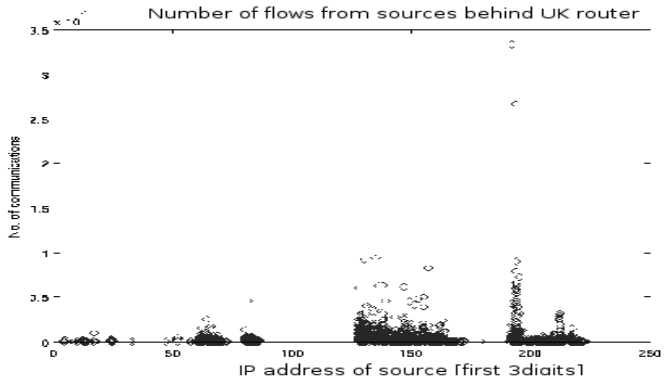

Figure 4.3: Number of flows from source hosts behind router UK1

The activities on the UK router have a much larger scale of volume and range of IP addresses as it is a major backbone router for GEANT and it is also the connection point to New York Point of Presence (PoP).

Figure 4.4 displays the number of flows that have been seen by the hosts behind the Hungary router [x-axis sorted by the first 3 digits of the IP address , from 0 to 255, covering the whole IP address spectrum, of the IP prefix]. It can be seen that there is IP address spoofing present in the Hungary domain. This can be observed both at source prefixes which have virtually covered the whole of IP address range.

It can also be observed that these hosts maybe creating attacks across the

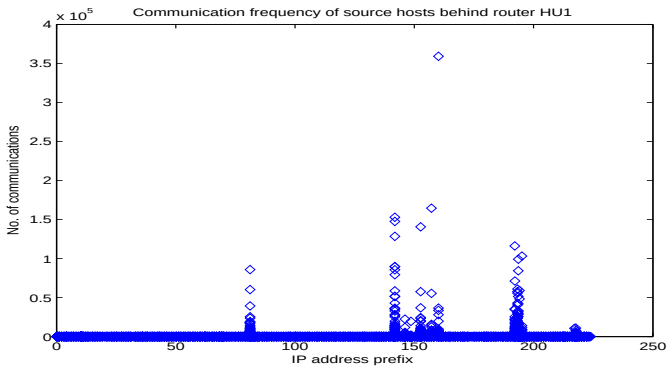

Figure 4.4: Number of flows from source hosts behind router HU1

internet by sending packets in large volumes all over the IP address range. It is only by real time monitoring that such anomalies can be detected and alarms can be raised to find the source of such activities which will compromise the performance of networks.

4.1.2 Packet rates

The following figures display the number of packets that have been sent out by the hosts behind some of the routers on GEANT. These figures are there to display the general trends and statistical behaviour of the hosts on a typical backbone network. *It must be noted that these statistics are subject to 1 in 1000 sampling rate.*

Figure 4.5 displays the Empirical Cumulative Distribution Function (CDF) of the packets sent out by the Slovakia router. Slovakia router is one of the smaller connection points on the GEANT network.

Figure 4.6 diplays the Empirical CDF of the aggregate number of packets sent out by the UK router in the day of measurement. That is the sum of all packets sent out by any one host behind the router to any destination worldwide. UK being one of the main connection points and the European-American exchange point has an order of magnitude more data across it than the relatively smaller routers such as Slovakia which was seen previously.

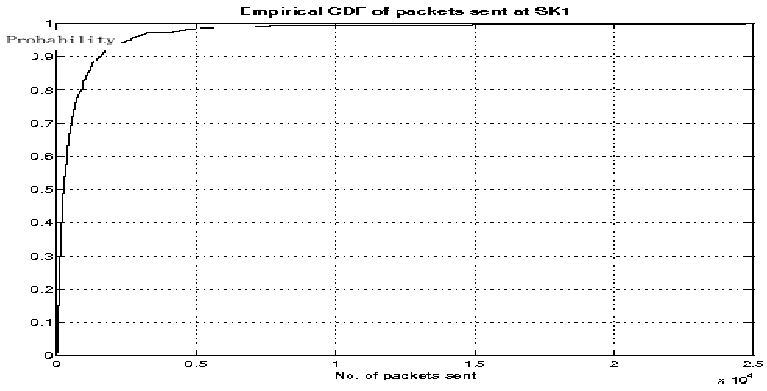

Figure 4.5: PDF of Packets sent by hosts behind router SK1

This trend was also observed when comparing the other routers together with the core routers such as Germany router (DE1) from which most of the GEANT traffic is traversed.

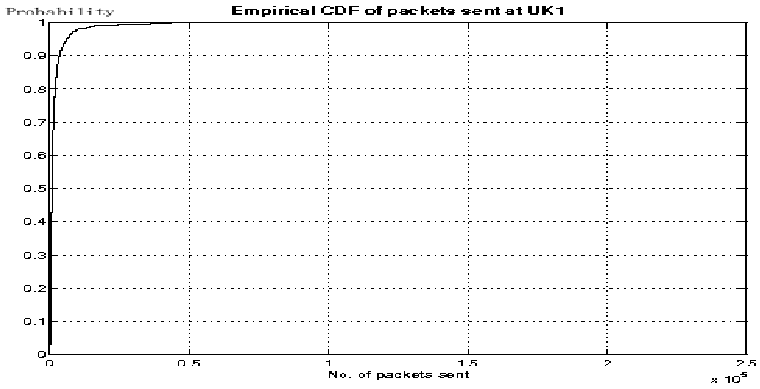

Figure 4.6: PDF of Packets sent by hosts behind router UK1

It can be seen that an extremely large number of hosts only send a very small number of flows, while a very small number of sources, which from their IP addresses could be identified as main university FTP servers and some file sharing facilities, initiate a large number of flows resulting to a high number of packets. This property conforms to the observations of Paxson et al. at [35].

4.1.3 Flow sizes distributions on GEANT routers

Even though the above figures are indicative of the *elephants* and *mice* nature of internet flows, a large number of flows (mice flows) only have a small number of packets, while very few flows (elephant flows) have a large number of packets [20]. It is interesting to view the size of flows transfered across networks by hosts and the frequency of occurrence of flows of different sizes. These results will enable network researchers to determine the type of traffic on networks. One of the hot topics looked at in network research has been the distribution of flow sizes on networks and the heavy detail In the Internet, heavy-tailed distributions have been observed in the context of traffic characterisation and in the context of topological properties. In the area of traffic characterisation, evidence indicates that Ethernet traffic exhibits self-similar properties [10]. It has been demonstrated that WAN traffic exhibits self-similar properties [11], as is the case for traffic specifically associated with WWW transfers The main implication of such discoveries is that most previous analytic work done in Internet studies adopted assumptions such as exponentially-distributed packet interarrivals. Conclusions reached under such exponentiality assumptions may be misleading or incorrect in the presence of heavy-tailed distributions.

the following graphs are demonstrative of such characteristics. They demonstrate the number of packets sent in flows, and the frequency of occurrence of each flows. It can be seen clearly that as the number of flows grows on bigger routers, the distribution of flow sizes tends to be spread across with the size of the flows gradually decreasing in frequency of their occurrence increases.

Figure 4.7 a displays the number of times a host has sent a packet of certain size, against the packet size on the x-axis. As there is not a great number of packets transmitted, the trends and variation of sizes can be clearly seen in this graph. Figure 4.8 displays the same property for a more

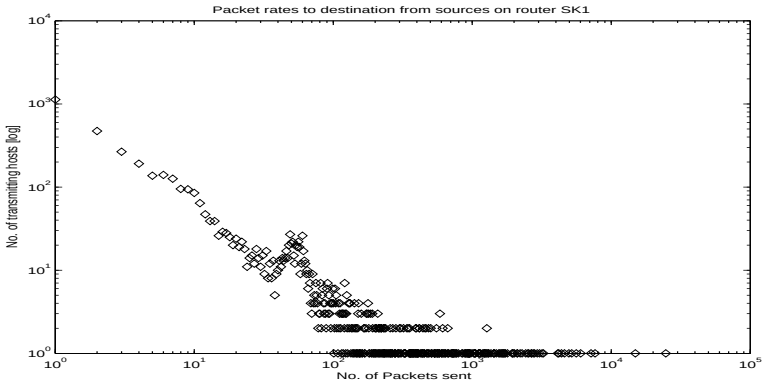

Figure 4.7: Activity rates of source hosts behind router SK1

busy router which is the German router, sitting at the core of the GEANT network. The German router, being at the core of the network, carries the most traffic on the GEANT network and it can be observed that the number of larger flows are much more on the DE1 router than the similar router at the UK.

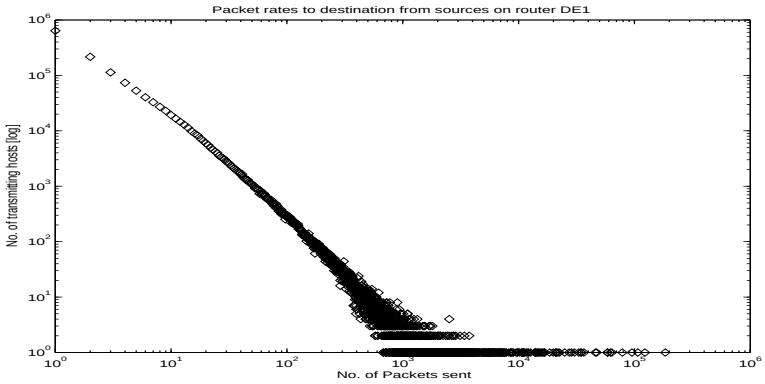

Figure 4.8: Activity rates of source hosts behind router DE1

The distribution is much smoother and without expanding the graph further via sampling for example it is not easy to observe any anomalies. This is due to the fact that there is much more human-generated traffic such as browsing and email on the UK routers than the core German router. This fact has increased the number of smaller flows on routers such as UK and Hungary and this can be observed by comparison to the German router.

4.2 Re-normalisation of Measured Usage

This section discusses re-normalisation of usage estimates from reports on uniformly sampled packets and how to take into account the effect of loss of reports in transit. , In an approach suggested by [32], each sampled value, such as a packet or flow length, must be re-normalised through division by its selection probability so as to obtain an unbiased estimator of the original. Let there be M packets in a given class, of which m are selected when sampling independently at rate p. We form an unbiased estimate $\widehat{M_1}$ of the total packets M in that class by $\widehat{M_1} = m/p$. Bytes are estimated similarly. Let the M original packets in the class have sizes $b_1, \ldots, b_M$ with total $B = \sum_{i=1}^{M} b_i$. Then $\widehat{B_1} = p^1 \sum_{i=1}^{'}, ..., M^{b_i}$ is an unbiased estimator of B, where $\sum^{'}$ denotes the random sum over sampled packets only.

An alternative re-normalisation uses the attained sampling rate, which may be calculated when sufficient information is provided in the measurements. In InMons sFlow [31], routers include the cumulative count of all packets arrived at the observation pointwhether sampled or notin each sampled packet report. By subtraction of counts, the collector can calculate the pool size, that is, the number of packets that arrived at the observation point between two given packets for which reports reached the collector. Let the pool size be N, of which n packets in all classes were sampled. The attained sampling rate for all traffic is n/N. Assuming this rate applies uniformly across all constituent classes, the estimate of the total packets in the class of interest is obtained by dividing the number m of sampled packet in the class by the attained sampling rate, yielding $\widehat{M_2} = mN/n$.

Analogous estimates for original bytes can be formed. In principle the attained sampling rate could be formed using either packet counts or byte counts. In practice, the estimate derived from packet counts is preferable: the estimate derived from byte counts has higher variance due to the variability of packet sizes. The attained loss rate between two received packets is

independent of which intervening packets were lost. Thus re-normalisation with the attained loss rate is less sensitive to deviations from independent sampling than re-normalisation with the target loss rate, provided that the deviations affect all traffic classes equally.

4.3 Variance of Usage Estimates

Assuming independent packet selection with probability p, $\widehat{M_1}$ has coefficient of variation $s_1 = ((1-p)/(pM))^{1/2}$, but $\widehat{M_2}$ offers some reduction in variance. Suppose that N,$M \to \infty$, with the proportion of packets M/N in the class under consideration converging to r. An application of the delta method [33] shows that the coefficient of variation of $\widehat{M_2}$ converges to $s_1\sqrt{1-r}$. The byte estimator $\widehat{B_1}$ has coefficient of variation $((1-p)\sum_{i=1}^{M} b_i^2/p)^{1/2}/B$. Since packet sizes are bounded above by the maximum transmission unit b_{max} of the link at which measurement takes place, it is possible to usefully bound this error above by $\sqrt{b_{max}/(pB)}$. Since the coefficients of variation are inversely proportional to the square root of the actual usage, larger contributions to usage are more reliably estimated than smaller ones. This is useful for applications. For example, the relative error in estimating high volume contributions to network usage is smaller than for general classes. This property was exploited by Jedwab, Phaal and Pinna [34] to identify heavy hitters through packet sampled usage. Their scheme assumes that only limited storage is available for ranking traffic classes by usage. If instantiating a new class would exceed storage capacity, the ranking information is truncated by discarding all classes except a certain number of the highest ranking. The probability of mis-ranking can be controlled to be small [32].

4.4 Uniform Sampling Probability

Usually packet sampling uses randomness in the sampling process to prevents synchronisation with any periodic patterns in the traffic. On average, 1 in every N packets is captured and analysed. As an example, for a flow of length between 1 to 1000 packets: p= 0.001.

4.5 Estimation method with increasing p

Given only a sample of values x1,...,xN from some larger population, many authors define the sample (or estimated) standard deviation by:

$$s = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{4.1}$$

The reason behind this equation is the fact that s^2 is an unbiased estimator for the variance σ^2 of the underlying population, if it is uncorrelated and has uniform variance of σ^2. However, s is not an unbiased estimator for the standard deviation σ; it tends to underestimate the population standard deviation. Although an unbiased estimator for "s" is known when the random variable is normally distributed which is not always true in case of network packet distributions as shown in works such as [11]. In Uniform sampling case, the relative standard deviation for unbiased estimation of the total packets n flows behaves roughly as $\sim 1/\sqrt{Np}$.

4.6 Estimating the number of active flows

Two definitions for counting flows: active flows and flow arrivals. A flow is active during a time period if it sends at least one packet during that time. Active flows with none of their packets sampled by the flow slicing process, will have no records; at least some of the flow records recorded should be

counted as more than one active flow, so that the total estimate will be unbiased. We count records with a packet counter c_s of 1 as $1/p$ flows and other records as 1 flow and this gives us unbiased estimates for the number of active flows:

$$\widehat{f} = \begin{cases} 1/p & \text{if } c_s = 1 \\ 1 & \text{if } c_s > 1 \end{cases}$$

Another way of expressing this range is as a percentage of the most likely value, in other words that the largest likely error is 4.8%. The following equation provides a simple estimate of the percentage error:

$$\%error \leq 196.\sqrt{\frac{1}{c}}$$

Plotting this equation in 4.9 shows how sampling accuracy improves as c increases.

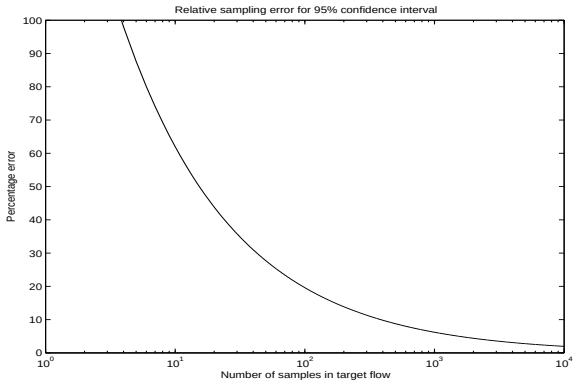

Figure 4.9: Relative sampling error for 95% confidence interval of missing a flow

It is observed from 4.9 that as the number of samples in the target flow increases, the relative error drops linearly and this is the basic conclusion of the effect of increasing the population in a sampling scenario.

4.7 Packet Count Estimator

The packet counter c_s in an entry is initialised to 1 when the first packet of the flow gets sampled, and it is incremented for all subsequent packets belonging to the flow. Let s be the number of packets in the flow at the input of the flow slicing algorithm. Equation below gives the formula for estimator $\widehat{s}$ for the number of packets in the flow:

$$\widehat{s} = 1/p - 1 + c_s$$

Lemma 1 $\widehat{s}$ as defined in is an unbiased estimator of s.

If $s = 1$, the only packet of the flow is sampled with probability p and in that case it is counted as $1/p - 1 + 1 = 1/p$ packets. With probability $1 - p$ it is not sampled (and it counts as 0). Thus:

$$E[\widehat{s}] = p \cdot 1/p + 0 = 1 = s.$$

If a router samples packets randomly with probability q before applying the flow slicing algorithm, the measurement system will want to estimate the number of packets S at the input of the packet sampling stage. Since $E[s] = qS$, it is easy to show that $\widehat{S} = 1/q\widehat{s}$ is an unbiased estimator for S.

4.8 Sparse Flows And Slicing

Consider an original flow with typical interpacket spacing τ . Suppose 1 in N packets are sampled from this stream. Typically, the interpacket spacing in the sampled stream is τN. If τN exceeds the flow interpacket timeout, then the original flow tends to decompose into a number of separate measured flows, depending on how the packets are bunched. The worst case is even spacing: each sampled packet would give rise to a separate measured flow. If τN is less than the flow interpacket timeout, the reverse holds, and the packets tend to be reported as a single measured flow. If evenly

spaced, a single measured flow would result; bunching may increase the number of measured flows. This presupposes there are multiple packets in the sampled stream. An original flow is called sparse, if sampling at a given rate typically yields more than one packet, with the typical interpacket time of the sampled packet exceeding the flow timeout. Thus an implicit and necessary condition for sparseness is that the typical flow length exceeds N. Packet size distributions of measured flows have previously been found to be heavy-tailed; see e.g. [35]. This leads to expectation of a noticeable number of long, and hence potentially sparse, flows. With sparse flows, packet sampling can then increase the number of measured flows and hence the downstream resource usage for those flows. It is not expected that sparseness to increase consumption of memory at the router. Whether or not splitting takes place, the original flow gives rise to at most one active measured flow at any time.

Peer to peer and streaming applications are candidates to produce sparse flows since they typically transmit packets over extended periods. Duffield et.al [27] call sparse those applications which may be expected to transmit sparse flows of traffic at typical sampling rates. "A single sparse original flow gives rise to multiple flow statistics. The increasing prevalence of longer file transfers by peer-to-peer applications, as much as 50% of traffic on some links, may lead to sparseness if the sampling rate is sufficiently low [27]". Ignoring sparseness can lead to substantial overestimation of the mean number of active flows, and hence the buffering resources needed to accommodate them in a router.

Duffield et.al in [27] conclude that flow slicing is more observed at moderate sampling rates 1 in N for $N = 10$ and $N = 100$, but declines once $N=$ 1000, since this exceeds the typical original flow length [27]. However in an analysis of CAIDA network in 1995, for a five minute trace, the average flow size for the IP flows was 10088 packets and 6,344,202 bytes [28]. The rise

in use of peer to peer applications has been rapid and recent other sparse applications may arise in future. This entails the need for a more dynamic sampling regime, in which the high volume of short flows, and long lasting flows of streaming and peer to peer applications are catered for.

In periodic sampling the router needs only decrement a counter. It has the potential disadvantage of introducing correlations into the sampling process: when a packet is selected, none of the following N-1 packets are selected. Although this does not bias against selection of any one packet, it can bias against selection of multiple packets from short flows.

4.9 Normalised recovery

Traditionally the recovery of flow characteristics have been by division by the sampling rate. For a sampling probability q and for sampled traffic $\alpha^{(q)}$ out of a sampling pool size N with inter-packet time-out τ_{th} it can be seen that normalised recovery is a trivial approach to inversion problem.

$$\alpha = \frac{1}{p}\alpha^{(p)} \tag{4.2}$$

where uniform random $(1/N)$ sampling is done:

$$\alpha = N\alpha^{(p)} \tag{4.3}$$

Below are the results of application of sampling in a simple network simulation scenario. These simulations are based on the statistics from the trace collected on 24 November 2004 on the GEANT network. In this simulation, 380000 flows across 23 routers and 100 nodes attached to them are generated. The statistics for these simulations are derived from the GEANT measurements discussed in the previous chapter, so they are subject to a bias already with short flows missing from them and a smoother distribution, due

to sampling and the central limit theorem, which states that assuming an infinite population with random sampling, the distribution of a sample average from a pool of data becomes like a Normal distribution as the sample size goes to infinity. The mean and the variance of the distribution will be the mean and the variance of of the original population divided by the sample size. Sampling and inversion process tends to ignore many short flows. This is also adversely true for the longer flows as this method of sampling tends to over estimate the number of the longer flow as discussed in section 4.8.

4.10 Sampling Rate and Missing Flows

The effects of sampling on long flows and the synthesis of *sparse* flows from long flows were discussed in section 4.8. In this section another effect of sampling is discussed which is the most important bias that sampling introduces in a trace file under analysis and that is the omission of many small packets which form small flows and are never sampled in the stream. Duffield et al. [27] mention that periodic sampling is very simple to implement: the router needs only decrement a counter. It has the potential disadvantage of introducing correlations into the sampling process: when a packet is selected, none of the following N-1 packets are selected. Although this does not bias against selection of any one packet, it can bias against selection of multiple packets from short flows. However, so far it has been assumed by other works that this effect would not be important for sampling from high speed links that carry many flows concurrently. In this case, successive packets of a given flow would be interspersed by many packets from other flows, effectively randomising the selection of packets from the given flow. While such randomisation may not be effective at lower speed routers carrying fewer flows (e.g. edge routers), packet sampling is not expected to be necessary for flow formation in this case.

This conclusion has been the basis of a series of experiments devised by the author. As many high speed links such as UKLight are there to serve for specific purposes such as large file transfers for databases of medical records or astrophysics simulation and measurements results, which will practically form one flow, in this case, despite the assumption made by the previously stated argument, there will be a bias against the smaller flows and the because there may not be too many active flows at any given time, flow slicing may not happen due to the fact that the router only needs to keep track of a limited number of flows and the continuous stream of data will avoid time-outs. In the next section two network scenarios are considered.

4.10.1 Scenario 1: Normal network characteristics

The simulator input for this case is the normalised CDF of packet statistics from GEANT on the measurement day as used previously. This simulation was based on 100 nodes located around 23 routers based on the CDF of locations of nodes behind GEANT routers. Around 300'000 flows were generated for his experiment, with the flow sources being based on the CDF of the transmission rates of the GEANT routers. The flow sizes were usually between 1 to about 250 packets per flow which is still small compared to the typical flows observed on routers today.

Figure 4.10 is a comparative view of the original normalised CDF with the normalised CDF of the sampled packet counts. The original non-sampled data set has a finer granularity however it is evident that the number of smaller flows in the CDF are lower after the sampling stage and this is due to the fact that many of the smaller flows are missed.

4.10.2 Scenario 2: Long flows, large packets

Todays networks are constantly evolving. Internet is no longer a simple web browsing and email medium. Peer to peer file sharing, audio stream-

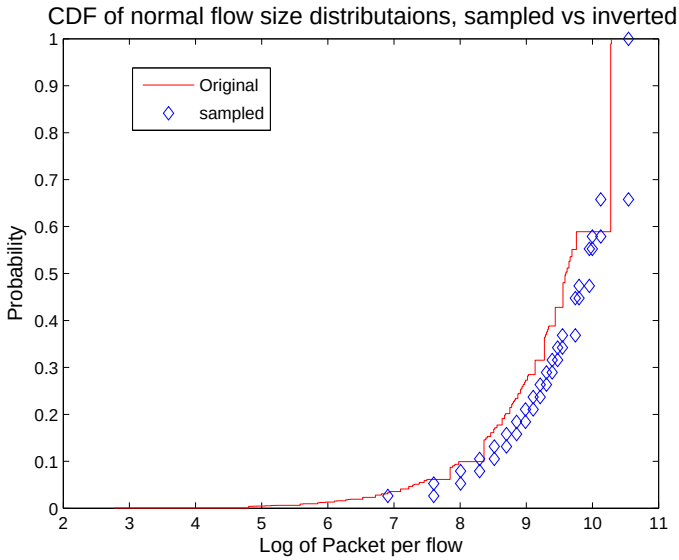

Figure 4.10: Comparison of the Normalised CDF of packet size distributions for flow sizes ranging from 1 to 244 packets per flow, no sampling (fine-grained) versus 1 in 1000 sampling (course-grained)

ing, torrents and online games have changed the intrinsic nature of flows on networks. many companies are connected via Virtual Private Network (VPN) and many send their traffic using the Multi-Protocol label Switching (MPLS) protocol. All of these mean that many individual flows between two remote sites of a company will look like a single large flow to the core router. In a network such as UKLight which has been designed to carry large GridFTP traffic and similar large flows of data transfers, it is vital that sampling is done In a way that the creation of sparse flows are minimised. In this section of the simulation, the range of packet sizes is increased from 1 to 244 previously to about 500-1200 packets per flow. The performance under various sampling methods are shown below.

Figure 4.11 displays the normalised Cumulative Distribution Function (CDF) of the total number of packets sent across the network of 23 routers based on their sizes, 500 to 1200 packets per flow versus the CDF extracted from the original data stream after sampling 1 in every 1000 is performed on

the packets and the inversion is achieved by multiplication by the sampling rate. In this context, since the minimum size of flows is about 500 packets, there will be the probability of missing some of the smaller flows, however as the simulator is designed in a way that the flows are sampled and generated sequentially, it may well be possible that the sampling is usual done far from the boundaries of the flows and somewhere in the middle of flows, so that all the flows are sampled. This is a limitation imposed by the simulation environment and in practical implementation it will not be present due to the fact that sampling will be done on a link that carries many flows concurrently so the packets of each flow will be interspersed by many other packets from other flows.

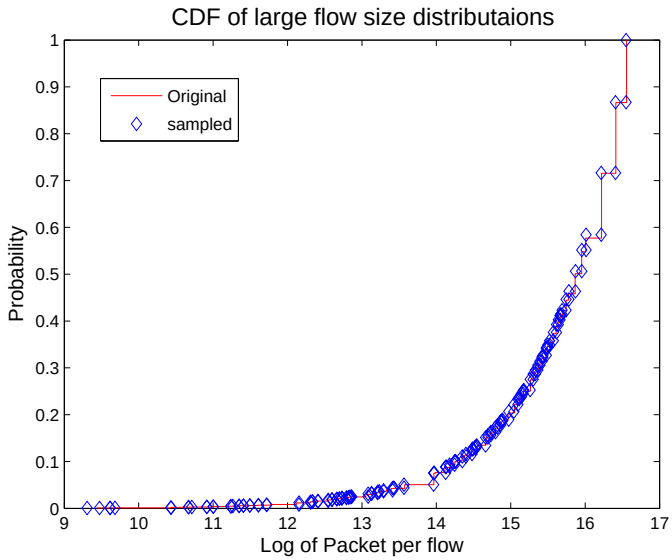

Figure 4.11: Comparison of the Normalised CDF of packet size distributions for flow sizes ranging from 500 to 1244 packets per flow, no sampling (fine-grained) versus 1 in 1000 sampling (course-grained)

It can be observed that especially due to the sequential nature of the sampling process, constant sampling has performed extremely well in this case and there is surprisingly no major difference between the shapes of the two CDF graphs. This is indicative that in an environment as such,

sampling 1 in a 1000 and then inverting the statistics using multiplications by the sampling rate is accurate enough for many purposes. However The great disadvantage of this method is that sparse flows can not be detected in such a scenario and in the next step of the work the author will focus on counting the number of flows as reported by the sampling process and compare it to the original stream under various sampling regimes.

It is interesting to view that there is a negligible error introduced by sampling with regards to the omission of the smaller flows. This is indicative that on a network link or topology where most of the flows are large flows comprising machine to machine flows such as file transfers and streaming applications, sampling at even a higher rate than 1 in a 1000 is possible but only providing that the router cache is not exhausted and flow splitting occurs as a result of freeing the tables for new flows or flow time-outs. In practice this can be tested on a large scale network such as PlantLab [38] and the author is in process of enabling such a facility for edge-based measurement and flow statistics aggregation.

4.11 Summary

In this section the affects of sampling on a typical traffic profile were discussed. It is observed that sampling has two distinct effects:

Under-estimation of short flows On a high speed link with many small flows such as web browsing, many of the smaller flows may not get sampled at all which makes the process of recovering the detailed recovery of original statistics difficult.

Creating sparse flows Sparse applications produce original flows comprising many packets with moderate interarrival times, that are likely to be split by packet sampling into multiple measured flows. Packet sampling increases the number of measured flows exported for ap-

plications such as peer-to-peer and streaming applications which are becoming more and more popular.

In the inversion problem, the interesting science is to infer the properties of the original traffic stream can be inferred from the packet sampled flow statistics. Previous work on these have not looked into details of statistics for different varieties of network scenarios. Duffield et al. [27] state that inference of total bytes and packets, possibly differentiated by flow key, is straightforward: dividing by the sampling rate the traffic rate represented in the measured flows yields an unbiased estimate of the originaltraffic rate and also show how to infer characteristics of the original traffic flows from the measured packet sampled flows. Whereas byte and packet volumes are estimated simply by dividing the measured quantities by the sampling rate, this approach does not work to estimate the number and mean length of flows, since some original flows will not be sampled at all. However the only disadvantage of this work is that they exploit the statistics of reported SYN packets for TCP flows. The limitations of this is that most of the anomalies, port scanning and even some major streaming applications do not use TCP. More difficult to infer are the detailed properties of the original flows: their arrival rate, their lengths. The main difficulty is that some flows may not be sampled at all; so it is not enough to simply form estimates through dividing the measured number of flows and their lengths by the sampling rate or look at TCP headers for SYN flag.

In analysis of simulation results, it was observed that when there are many short flows are present, the sampling schemes currently in use in internet can fall short of producing detailed statistics. Many of the short flows are missed with periodic and constant sampling schemes. However the statistics are not always too biased for the longer flows and they can be realistic. in terms of sparse flows as a result of slicing larger flows to smaller ones, work has to be done in the simulation and in practice to analyse the

evidence that the existence of such flows greatly misleads the statistical analysis of network traffic characterisation.

Chapter 5

Sample and Export in Routers

In this section we look at a more detailed analysis of the effect of sampling as performed by netflow on higher order statistics of the packet and flow size distributions. For the analysis of packet sampling application is used by NetFlow, we emulated the NetFlow operation on a 1 hour OC-48 trace, collected from the CAIDA link on 24^{th} of April 2003, from 8:00 tp 9:00. This data set is available from the public repository at CAIDA [28]. The trace comprises of 84579462 packets with anonymised source and destination IP addresses. An important factor to rememberer in this work is the fact that the memory constraint on the router has been relaxed in generating the flows from the sampled stream. This means that there maybe more than tens of thousands of flow keys present at the memory at a given time, while in NetFlow, the export mechanism empties the buffer list regularly which can have a more severe impact on the resultant distribution of flow rates and statistics[3].

[3]The processing of the data was done using tools which are made available to the public by the authors.

5.1 Effects of the short time-out imposed by memory constraints

As observed in table 5.1, the mean does not have a great variation, possibly because distributions of packet sizes within single flows do not exhibit high variability. The standard deviation of the estimated data rate is higher than the corresponding standard deviation for the unsampled data stream. In the absence of any additional knowledge about the higher level protocol, or the nature of the session level activity, in the unsampled data stream, each flow can be thought of as having packets of varying sizes that are more or less independent from one another. Thus, the whole traffic profile results from the addition of many independent random variables which, by the central limit theorem, tend to balance among themselves to produce a more predictable, homogeneous traffic aggregate. However, simple inversion eliminates this multiplicity of randomly distributed values by introducing a very strong correlation effect, whereby the size of all the packets in a reconstructed flow depend on the size of a very small set of sampled packets. This eliminates the possibility for balancing and thus increases the variability of the resulting stream, i.e. its standard deviation.

However, the skewness and kurtosis do change. Skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable. Roughly speaking, a distribution has positive skew (right-skewed) if the right (higher value) tail is longer and negative skew (left-skewed) if the left (lower value) tail is longer (confusing the two is a common error). Skewness, the third standardised moment, is written as γ_1 and defined as:

$$\gamma_1 = \frac{\mu_3}{\sigma^3}$$

where μ_3 is the third moment about the mean and σ is the standard deviation.

Kurtosis is more commonly defined as the fourth cumulant divided by

the square of the variance of the probability distribution,

$$\gamma_2 = \frac{\kappa_4}{\kappa_2^2} = \frac{\mu_4}{\sigma^4} - 3$$

which is known as excess kurtosis. The "minus 3" at the end of this formula is often explained as a correction to make the kurtosis of the normal distribution equal to zero. The skewness is a sort of measure of the asymmetry of the distribution function. The kurtosis measures the flatness of the distribution function compared to what would be expected from a Gaussian distribution. Table 5.1 illustrates the data rates $d(t)$ per interval of measurement. Inverted data rates, by dividing $d(t)$ by the sampling probability q, are shown as $dn(t)$.

Table 5.1: The statistical properties on Data rates $d(t)$

Dataset,bin(secs)	Mean	STD	Skewness	Kurtosis
$d(t)$, 30	3.6019e+08	2.2274e+07	0.5421	0.6163
$d_n(t)$, 30	3.5919e+08	2.9109e+07	0.3837	0.4444
$d(n) - d_n(t)$, 30	*1.0009e+06*	*1.6748e+07*	*-0.2083*	*0.7172*
$d(t)$, 120	1.4408e+09	7.8650e+07	0.7398	1.6190
$dn(t)$, 120	1.4368e+09	9.5216e+07	0.3274	0.9268
$d(t) - d_n(t)$, 120	*4.0037e+06*	*3.7652e+07*	*-0.2971*	*-1.1848*
$d(t)$, 300	3.6019e+09	1.8491e+08	1.3058	3.7451
$d_n(t)$, 300	3.5919e+09	2.1248e+08	1.1016	2.5408
$d(t) - d_n(t)$, 300	*1.0009e+07*	*6.1039e+07*	*0.1840*	*-1.1628*

Table 5.2 illustrates the packet rates $p(t)$ per interval of measurement. Inverted packet rates, by dividing $p(t)$ by the sampling probability q, are shown as $pn(t)$. The distributions before and after sampling are extremely close, and thus their difference tends to exaggerate those small difference that they do have. That is the reason of the enormous skewness and kurtosis that are observed. The skewness of the reconstructed stream is smaller than that of the unsampled stream this means that the reconstructed distribution is more symmetric, that is , it tends to diverge in a more homogeneous

manner around the mean. Additionally, it is positive, meaning that in both cases the distribution tends to have longer tails towards large packets rather than towards short packets, concentrating its bulk on the smaller packets. If we conclude that small flows (flows consisting of a small number of packets) tend to contain small packets, then it is clear that this smaller packets will be underrepresented and the distribution will shift its weight towards bigger packets (members of bigger flows). Thus, it will become more symmetric and hence less skewed.

Table 5.2: The statistical properties on Packet rates $p(t)$

Dataset,bin(secs)	Mean	STD	Skewness	Kurtosis
$p(t)$, 30	7.0483e+05	3.1162e+04	-0.4007	0.7415
$p_n(t)$, 30	7.0483e+05	3.1359e+04	-0.3584	0.6072
$p(t) - p_n(t)$, 30	*-5.7333*	*5.4148e+03*	*9.1469*	*96.0659*
p(t), 120	2.8193e+06	1.1215e+05	-0.3875	1.2027
pn(t), 120	2.8193e+06	1.1178e+05	-0.3759	1.2238
$p(t) - p_n(t)$, 120	*-22.9333*	*3.0157e+03*	*4.7140*	*26.1079*
p(t), 300	7.0482e+06	2.5128e+05	0.1305	1.6495
pn(t), 300	7.0483e+06	2.5152e+05	0.1433	1.6597
$p(t) - p_n(t)$, 300	*-57.3333*	*2.1047e+03*	*2.4298*	*8.9377*

The Kurtosis decreases in all of the considered examples. This means that the reconstructed streams are more homogeneous and less prone to outliers when compared with the original traces. Thus, more of the variance in the original traces in packet size can be attributed to infrequent packets that have inordinately big packets that were missed in the sampling process, and thus the variance in the reconstructed stream consists more of homogeneous differences and not large outliers. However, both the reconstructed and unsampled streams tend to have long, heavy tails.

5.1.1 The two-sample KS test

The two-sample Kolmogorov-Smirnov test is one of the most useful and general non-parametric methods for comparing two samples, as it is sensitive to differences in both location and shape of the empirical cumulative distribution functions of the two samples. A CDF was calculated for the number of packets per flow and the number of octets per flow for each of the 120 sampling intervals of 30 seconds each, both for the sampled/inverted and unsampled streams. Then, a Two-Sample Kolmogorov-Smirnov Test with 5% significance level was performed between the 120 unsampled and the 120 sampled and inverted distributions. In every case the distributions before and after sampling and inversion were found to be significantly different, and thus it is very clear that the sampling and inversion process significantly distorts the actual flow behaviour of the network.

5.2 Practical Implications of Sampling

The effects of sampling on network traffic statistics can be measured from different perspectives. In this section we will cover the theories behind the sampling strategy and use some real data captures from CAIDA in an emulation approach to demonstrate the performance constraints of systematic sampling.

5.2.1 Inversion errors on sampled statistics

The great advantage of sampling is the fact that the first order statistics do not show much variation when the sampling is done at consistent intervals and from a large pool of data. This enables the network monitoring to use the sampled statistics to form a relatively good measure of the aggregate measure of network performance. Figure 5.1 displays the data rates $d(t)$, in number of bytes seen per 30 second interval, on the one hour trace.

The inverted data $d(t)$ is also shown with diamond notation, showing the statistics gathered after the sampled data is multiplied by the sampling rate. The black dots display the relative error per interval, $e(t) = \frac{d(t)-dn(t)}{d(t)}$.

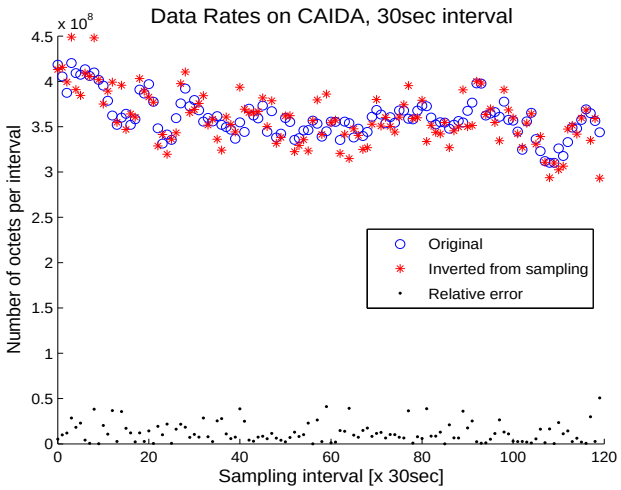

Figure 5.1: Data rates per 30 second interval, original versus normal inversion of sampled

Figure 5.2 displays the packet rates $p(t)$, the number of packets per 30 second interval, versus the sampled and inverted packet rates $pn(t)$. In this figure, it can be observed that the inversion does a very good job at nearly all times and the relative error is negligible. This is a characteristics of systematic sampling and is due to the central limit theorem.

It can be readily seen that the recovery of packet rates by simple inversion is much better than the recovery of data rates. This is because sampling one in a thousand packets deterministically can be trivially inverted by multiplying by the sampling rate (1000): we focus on packet level measurement, as opposed to a flow level measurement. If the whole traffic flow is collapsed into a single link, then if we sample one packet out every thousand and then multiply that by the sampling rate, we will get the total number of packets in that time window. We believe that the small differences that

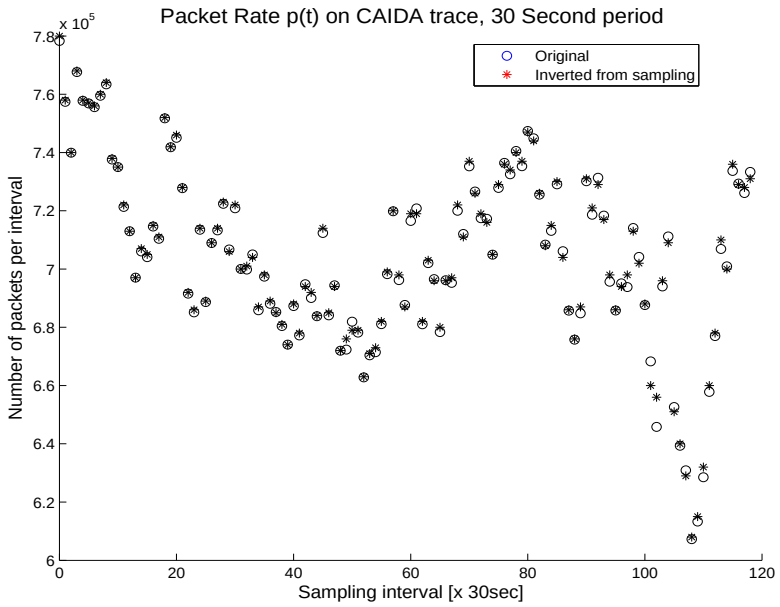

Figure 5.2: Packet rates per 30 second interval, original vs inversion of sampled

we can see in Figure 2 are due to the fact that at the end of the window some packets are lost (because their 'representative' was not sampled) or overcounted (a 'representative' for 1000 packets was sampled but the time interval finished before they had passed). We believe these errors happen between measurement windows in time, i.e. they are window-edge effects.

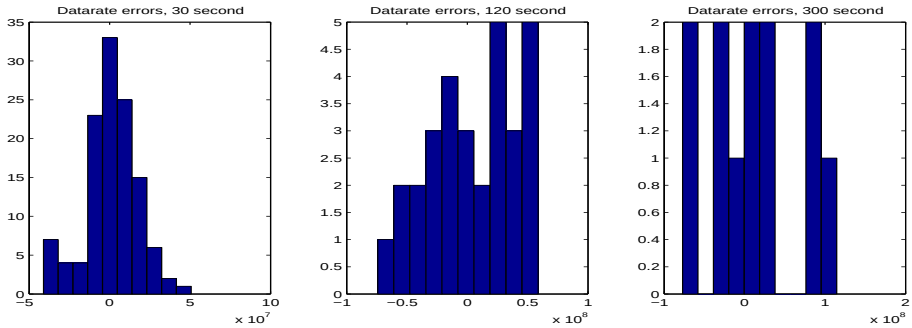

Figure 5.3: Standard Sampling & inversion error on data rates, different measurement bins

The inversion property described above does not hold for measuring the

number of bytes in a sampling interval. Simple inversion essentially assumes that all packets in a given flow are the same size, and of course this assumption is incorrect. It is to be expected that the greater the standard deviation of packet size over an individual flow, the more inaccurate the recovery by simple inversion will be regarding the number of bytes per measurement interval. Figure 5.4 displays the standard error rate on packet rate recovery in different measurement intervals.

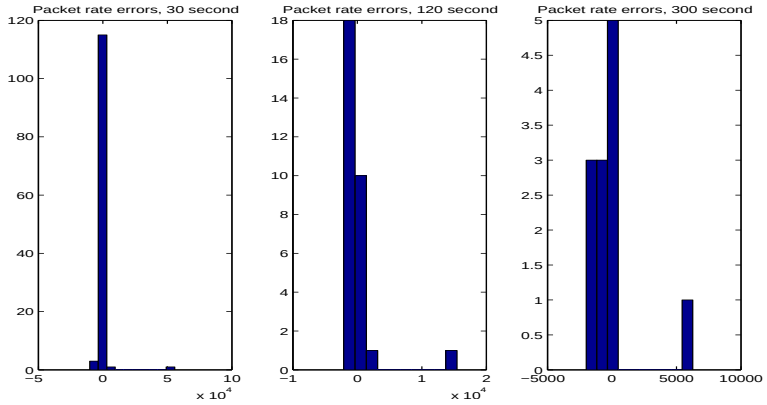

Figure 5.4: Sampling & inversion error on packet rates, different measurement bins

5.2.2 Flow size and packet size distributions

Figure 5.5, displays the CDF of packet size distribution in all the flows formed from the sampled and unsampled streams. The little variation in the packet size distribution conforms to the findings of the previous section where it was discussed that the packet sampling has low impact on the packet size distribution. Due to the fact that the flows tend to be densely concentrated at the higher end towards the smaller flows, all the following graphs are displayed on a log scale.

Figure 5.6:1 shows the effect that the distribution of packet lengths can have on the distribution of flow lengths when periodic packet sampling is applied. As flows reconstructed from a sampled packet stream are predom-

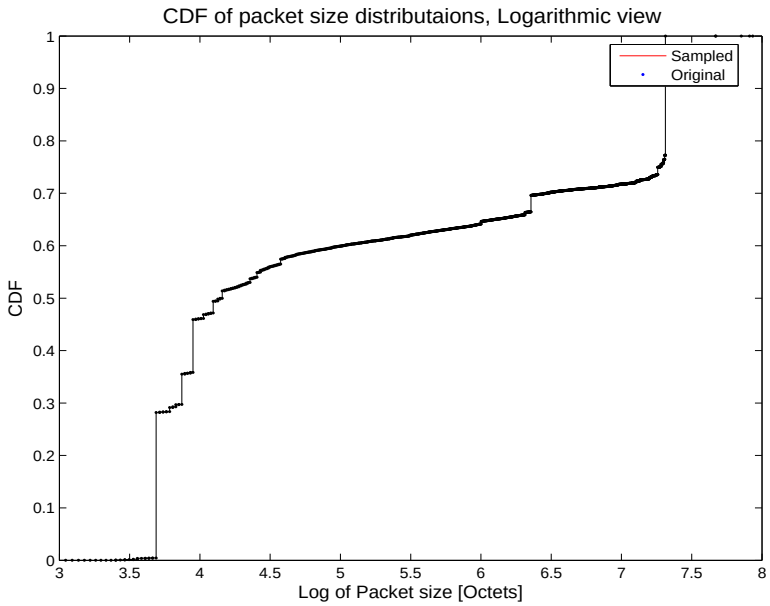

Figure 5.5: Normalised CDF of packets distributions per flow, original vs inverted

inantly formed by just one packet, their length distribution follows that of single packets (Figure 5.5). That is the reason for the sharp jump near 1500 octets, as this characteristic originates from the maximum frame size in ethernet networks.

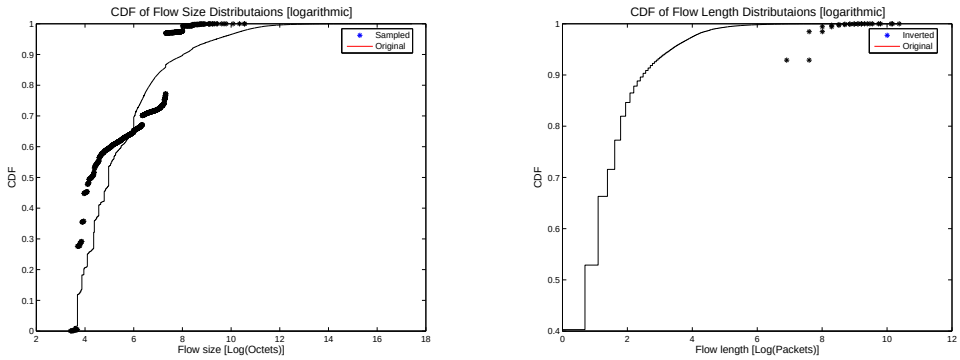

Figure 5.6: Normalised CDF of flow size in packets [figure] & length in bytes [right] per flow, original vs inverted

From Figure 5.6:2 , it can be readily seen that, in the sampled stream, more than 90 percent of flows consist of a single packet, whereas in the un-

sampled case a much grater diversity in flow lengths exists for small flows. This is due to the fact that simple packet-based deterministic sampling under-represents short flows, and those short flows that are indeed detected by the procedure after sampling usually consist of a single packet. Thus, short flows are either lost or recovered as single packet flows, and long flows have their lengths reduced.

Chapter 6

Inference of Network Flow Statistics

It is inevitable that sampling, as done today on the core routers of the networks and the aggregation points, is not optimal. The short falls are in two distinct regimes, the very short flows, which typically comprise of web page look ups, emails and may be even form part of Denial of Service (DoS) attacks and anomalies, are mis-represented in the aggregate statictics. On the other hand, at presence of many large flows at a router which carries thousands of flows, Netflow has a limited amount of fast memory available to it which must use efficiently to gather summaries for all the flows. This leads to premature termination of flow statistic collection for large flows that have been on the table for a long time and have reached the buttom of the list. In such circumstances, a single flow which may be part of an audio or video stream or a peer to peer file transfer stream, is split into many smaller flows and each flows is accounted for and output separately. This leads to over presentation of larger flow counts on the aggregated statistics reported by Netflow or the alternative software present at the core router.

6.1 Adaptive sampling

Adaptive sampling has been suggested under different names previously. Duffield et al. in [26] suggests a deterministic adaptive sampling to be applied on *flows*. The sampling strategy described [26] is currently used to collect NetFlow records collected extensively from a large IP backbone. The collection infrastructure deals with millions of NetFlow records per second. Measurements from some tens of routers are transmitted to a smaller number of distributed mediation servers. The servers aggregate the records in multiple dynamic ways, serving the needs of a variety of network management applications. The aggregates are sent to a central collection point, for report generation, archival, etc.

The major difference between the above work and the proposed work in this thesis is that in the above work, two stages of sampling are present as shown in figure 6.1.

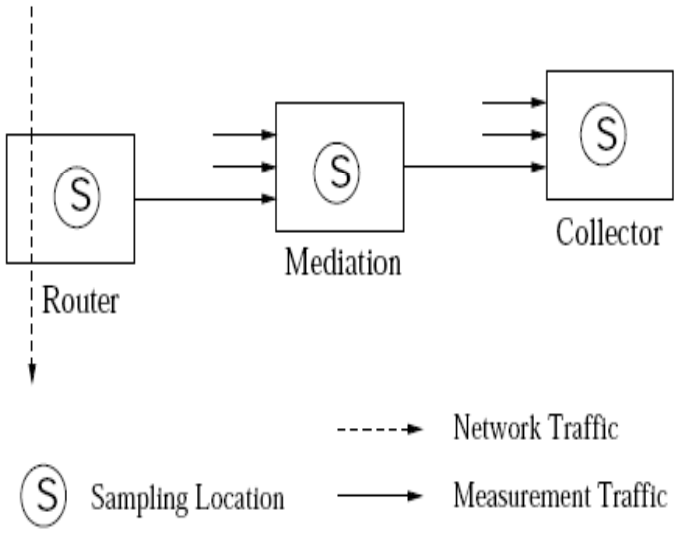

Figure 6.1: Flow of measurement traffic and potential sampling points: (left to right) packet capture at router; flow formation and export; staging at mediation station; measurement collector. [Figure courtesy of AT & T]

In stage one, packet sampling has been performed on the router. It is however assumed that the sampling has an un-biased estimator:

The aim is to obtain an estimate of X from a subset of sampled values, and, generally, without needing to know the original number n of sizes. $\widehat{X}$ is said to be an unbiased estimator of X if $\mathrm{E}\widehat{X} = X$. $\widehat{X}$ is unbiased if

$$X = \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} p(x_i) r(x_i) = E\widehat{X}$$

This happens for all collections $\{x_i\}$ if and only if

$$r(x) = x/p(x) \text{ for all } x$$

And this assumption has been held throughout the report.

if the sampling rate is 1 in N, based on the average number of packets per flow in a given interval τ, one can predict the traffic characteristics and set the sampling rate accordingly to capture more small flows, or to increase the time-out in order to get full-lengths of large flows. Also, simulation has been modified to allow end nodes capture their own packet trace and store them., this can be used for the prediction model.

The optimum choice of sampling on a link where adaptive sampling is utilised is sFlow [31]. There has been work on improving netflow however this requires periodic updates to the software at a router running netflow [39]. Sampling tools such as sFlow sample packets without building up flow records from them. This sampled substream can then be passed to an aggregation point in order to generate the flow statistics. The great advantage of these types of solutions is that they report full packet headers together with a portion of the packet payload and this provides much richer raw data for analysis. The disadvantage discussed by Estan et al. [39] is that they do not benefit of the compression achieved by flow records that count

more than one packet. However on a high speed link this may more than compensate for changing hardware and software configurations of a NetFlow router.

Duffield et al. [27] suggest that it may be advantageous to adjust flow delineation criteria with sampling rate in order to match the flow definition to the underlying nature of the transactions that generate the traffic. One case that they investigate is scaling the interpacket timeout inversely with the sampling rate in order to capture longer lived packet streams as a single flow whether the volumes of flow statistics, and the number of active flows, can be easily predicted. Packet traces are not available at most points in a network; heterogeneity of traffic prevents generalising the analysis from a given trace to arbitrary network sites.

The short fall of method suggested at [26] is use of sampled flow statistics. Given a set of statistics of unsampled flows perhaps derived directly from flow measurements the model predicts the flow export rate and mean number of active flows that would result if instead flow statistics from a sampled version of the original packet stream were formed.

The major step to be taken in the work by the author is to analyse the sampling process used by a tool such as sFlow on a stream of real traffic trace, ideally such traces can be collected from different networks where dominant applications maybe smaller flows such as browsing and emails to those with larger transfers such as peer to peer data and streaming applications.

6.2 Network Tomography Using Distributed Measurement

Enterprise and ISP network management requires thorough knowledge of network status, malfunctions in the routers, firewall configurations, link utilisation, delay and time out measures, traffic matrices and etc. A slight

mis-configuration in a firewall or IP address assignment of a NAT device is enough to cause two devices not being able to communicate with each other. Figure 6.2 displays an extremely simple scenario of an enterprise network over three offices worldwide, where the two smaller centres connect to the main servers and facilities in the head office over internet using a technology such as VPN as an example. It is possible that Node A is able to talk to the mail exchange server located at the main head quarters but Node B is not able to connect to the same server due to a mis-configured firewall. In this case, the optimum placement of measurement and monitoring points at the core and edges of the enterprize network allow the system to narrow down its search to the possible causes of such a scenario.

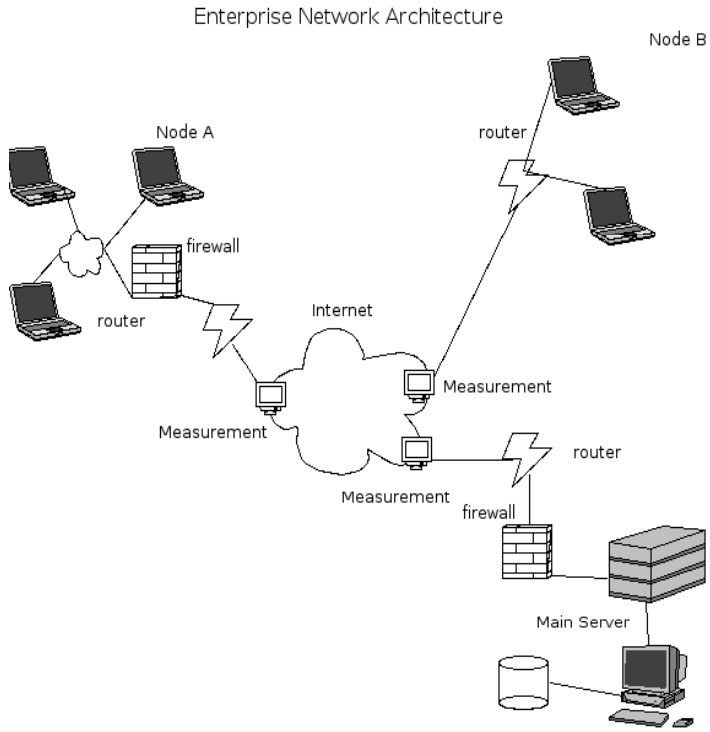

Figure 6.2: Typical architecture of an Enterprise network, various laptops all around the world are connecting to the enterprise server via VPN configuration. It is natural for them to have to go through many firewalls and proxy servers in order to connect to the desired end point.

There has been some work done in this fields recently and it is becoming a topic of interest amongst network researchers however the field is still very immature. Passive monitoring of IP flows at multiple locations in a network has not been possible in the past due to many privacy and inter connection issues. Router vendors softwares usually would not report rich traffic sets and the SNMP reports of the core routers are not rich enough to enable real inference of traffic flow characteristics. However recently there has been networks such as PlanetLab and GEANT which have lent themselves into measurement projects and give out (usually anonymised) traffic statistics reports such as NetFlow, ISIS and BGP data.

The common objective of such a distributed monitoring system is to sample packets belonging to a large fraction of IP flows in a cost-effective manner by carefully placing monitors and controlling their sampling rates. In recent work by K.Suh et al. [41], they consider the problem of where to place monitors within the network and how to control their sampling. To address the tradeoff between monitoring cost and monitoring coverage, minimum cost and maximum coverage problems under various budget constraints are looked into and it is shown that all of the defined problems are NP-hard.

Inference of such information leads to being able to perform passive tomography of the link failure information, whilst most of the work done in the tomography area thus far relies on actively injecting traffic on the network which may not always be feasible due to injection of biased traffic. Another disadvantage is the fact that it is not possible to perform the techniques on already collected traffic traces. An example of a scenario when the tomography can lead to discovery of a mis- configured firewall is demonstrated in figure 6.3. In this figure hosts A, B, C and D form a ring network. However on the route from node C to node B there is a firewall which has been recently re configured by an operator which has closed connection possibility

from *node C* to *node B*.

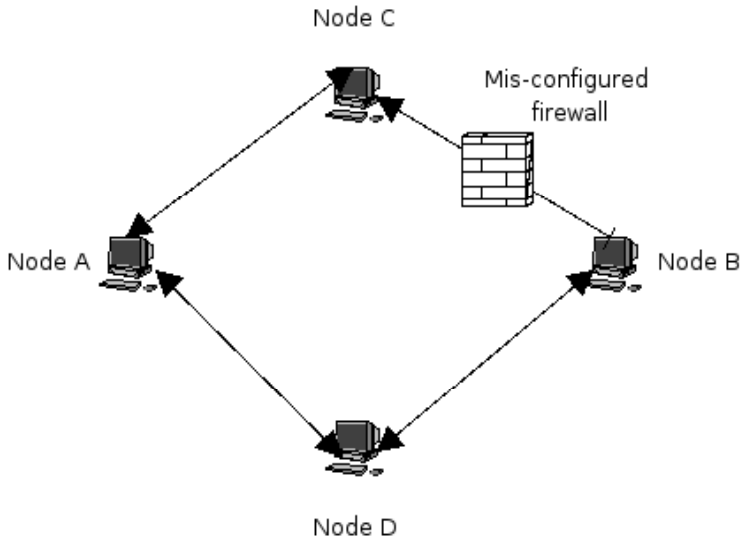

Figure 6.3: A simple double ring network, nodes are connected bi-directionally but there is a mis-configured firewall which stops packets from*node C* to *node B*.

It can be seen that in presence of explicit routing policy, some packets may not make it from C to B, this can even include acknowledgement packets (ACK) which are sent after a message was successfully received at C. In the presence of Equal Cost Multi Path (ECMP) this has even a more sever form, as can be seen in picture 6.4. The communication process happens successfully between the two nodes and it just takes another path. So it becomes inevitably more difficult to trace the cause of failure of the group of packets that have been missing from the flow and this may be a very small fraction of the traffic in case of traces collected at the edge of a large enterprise network.

The major disadvantage of novel work done in [41] is the fact that it considers IP networks in which each IP flow is routed along a single path. Because of single path routing, it is possible to observe all packets in the flow by monitoring any one of the links on the flow's path. However even

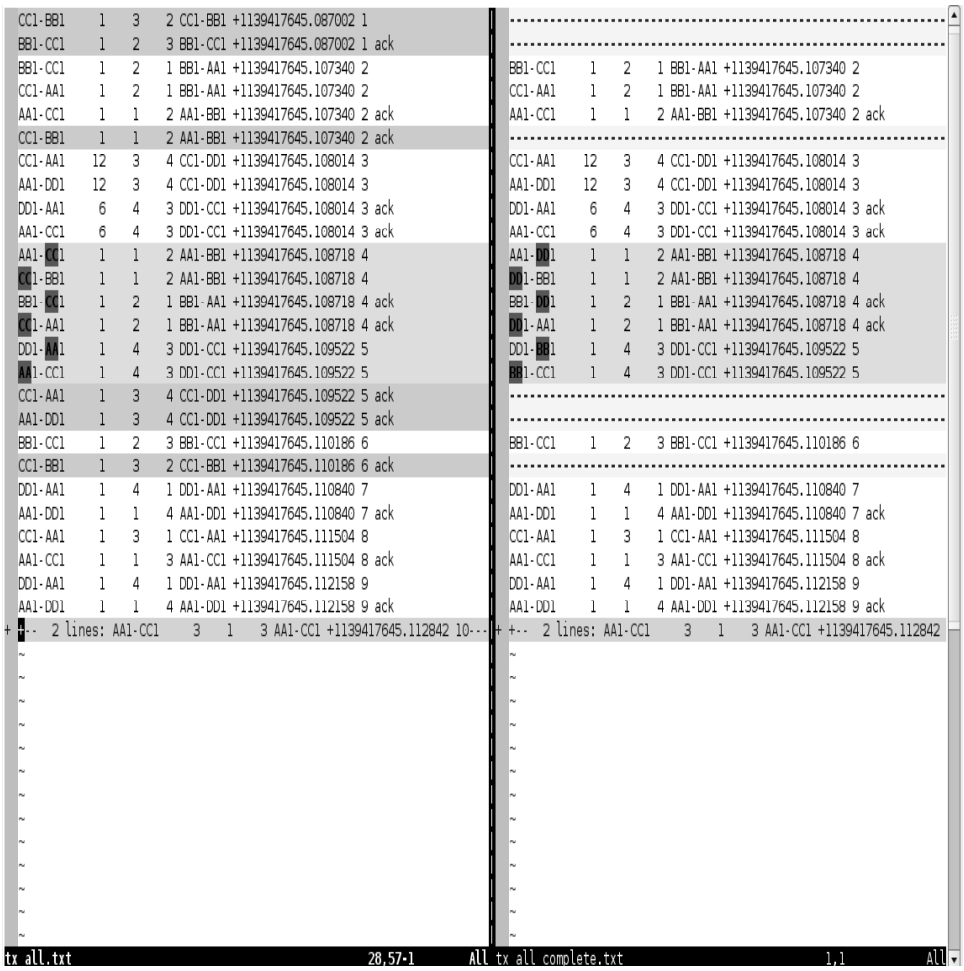

Figure 6.4: A small traffic flow between nodes. Packets from *node C* to *node B* can take alternative route to get to destination hence avoiding the firewall. This makes the task of finding failure points in the network more difficult.

with such assumption it is shown that the problems are NP-hard.

The vital decision at such a problem is the location of the measurement points. There are two places where measurement can place. Complete traffic traces can usually be collected at the edge devices, and sampled statistics can be output by the core routers.

The solution to a distributed monitoring problem consists of two parts:

1. Set of end notes and links at which to place a monitor and being able to query them.

2. Measurement and querying strategy (e.g. sampling rate and query frequency) at each monitoring place.

If the end to end query and response is considered and it is possible to query all the end nodes and all the routers in between them, the above example will have the following criteria.

Suppose there are i alternative routes to take between n nodes, the number of possible of possible routes between them will be ni. In the above example, the routes will be as followed:

1. $node_A, Link_{AC}, Link_{CB}, node_B, Link_{BC}, link_{CA}, node_B$

2. $node_A, Link_{AC}, Link_{CB}, node_B, Link_{BD}, link_{DA}, node_B$

3. $node_A, Link_{AD}, Link_{DB}, node_B, Link_{BC}, link_{CA}, node_B$

4. $node_A, Link_{AD}, Link_{DB}, node_B, Link_{BD}, link_{DA}, node_B$

The most simple way to perform the query at such a scenario will be to perform a binary search on each iteration of the transaction process in order to find out the missing traffic packets before or after the point of failure.

```
    function binarySearch(a, value, left, right)
while left  right
mid := floor((right-left)/2)+left
```

```
if value > a[mid]
left := mid+1
else if value < a[mid]
right := mid-1
else
return mid
return not found
```

The tomography of the network by passive measurement in such a way has not been experienced before to the best of author's knowledge and this is certainly an interesting area for exploiting the measurement and adaptive sampling opportunities.

Chapter 7

Conclusions and Future Plans

In this report the consequences of collecting packet sampled flow statistics are examined. It is pointed out that the flows in the original stream whose length is greater than the sampling period tend to give rise to multiple flow reports when the interpacket time in the sampled stream exceeds the flow timeout. In practice this occurs predominantly for traffic generated by peer-to-peer applications. Such traffic is on the rise, motivating the need to better understand the implications for resource usage in the measurement infrastructure of such splitting.

The aim of this work is to enable prediction of original flow rates and the number of active flows from sampled flow traffic statistics. It is possible to predict a coarse statistics on flow details, however the information which are of interest is the exact numbers of smaller flows which may be missed by sampling at high intervals, or even more importantly for the high capacity networks that are the end target of this work, it is to be able to keep track of the larger flows without the limitation put in place by sampling schemes today, which usually time-out a flow after 15 seconds of seeing the last packet in a flow, or 30 seconds after creation of the flow entry. As also

noted by Duffield et.al [26], failing to take account of sparse flows (those vulnerable to splitting) can lead to underestimation of the total flows, and severe overestimation of the size of the buffer needed to accommodate active flows. Even though the sampling rates are configurable, they have to be programmed into a NetFlow router and that is part of the work done previously in this field. This is indicative that in an environment with comparatively large flows, sampling 1 in a 1000 and then inverting the statistics using multiplications by the sampling rate is accurate enough for many purposes. However The great disadvantage of this method is that sparse flows can not be detected in such a scenario and in the next step of the work the author will focus on counting the number of flows as reported by the sampling process and compare it to the original stream under various sampling regimes. For practical implementation, The authors aim is to use the sFlow module within CoMo to have the ability of sampling adaptively.

Nowadays networks are constantly under attack from Distributed Denial of Service (DDoS) attacks, port scanning applications and worms which mostly contain a high number of SYN packets generating a new flow record which may or may not be sampled by the router simply using NetFlow 1 in 1000 sampling and many of them will be missed. Hence many attack detection research projects have to rely on packet trace collections over periods of time to observe such anomalies and this is certainly not viable for a large ISP network trying to manage the network in real time in terms of provisioning and QoS constraint maintenance.

While sampling can be compensated for in reports that measure the traffic in packets or bytes it has been proven that it is impossible to measure traffic in flows without bias. To date there has been no work on recovering detailed properties of the original unsampled packet stream, such as the number and lengths of flows. Duffield et al. at [27] suggest It may be advantageous to adjust flow delineation criteria with sampling rate in order

to match the flow definition to the underlying nature of the transactions that generate the traffic. This is the basic introduction to adaptive sampling which will be the focus of the next stage of this work.

7.1 Plans for the next stage of the PhD research

It is proposed to look into an extension to the sampling work and the inversion problem which entails the use of more detailed statistics such as port numbers and TCP flags in order to be able to infer the original characteristics from the probability distribution functions of such variables. This will enable a more detailed recovery of original packet and data rates for different applications. The inference of such probabilities, plus use of methods such as Bayesian inference, would enable a forecasting method which would enable the inversion of the sampled stream in near real time.

In this work, multiple information, such as port number and distribution of packets per port, source and destination IP addresses, and information readily available such as packet counts from SNMP reports are gathered in order to enable the creation of entropy tables, which will give more detailed about the characteristics of the flows and aid in the inversion of the statistics.

In related work, we will be looking at adaptive sampling schemes, looking at techniques replacing the NetFlow, such as InMon's sFlow [31], in order to use the forecast statistics to change of the sampling rate on-the-fly, which would accommodate for the changes in the traffic profile.

In tomography experiments, the generation of traffic-load, topology and applications scenarios will be looked at. The aim is to create a tool which can simulate a network to resemble a particular network, focusing on Corporate networks. The objective is to provide a validated tool for analysing failures and changes of characteristics of networks such as introduction of new sites or link failures. Another objective is to be able to pin point the failure points in a network by optimally adjusting the number of monitors, their locations

and their sampling rates. The placement and number of monitoring points is showed to be an NP-hard problem [42]. However the heuristics will lead to an exciting new range of opportunities for research into network management.

Bibliography

[1] Case for Support: L Sacks, S Bhatti, D Parish, I Philips, A Moore, R Gibbens, I Pratt, I Graham, `http://masts.lboro.ac.uk/`

[2] A Grid for Particle Physics - Managing the Unmanageable, D. Britton, A.T.Doyle, S.L.Lloyd, UK e-Science All Hands Conference, Nottingham, September 2004.

[3] An Introduction to NetherLight: `http://www.netherlight.net`

[4] Estan and G. Varghese, "New directions in traffic measurement and accounting" in Proceedings of the 2001 ACM SIGCOMM Internet Measurement Workshop, pp. 75–80, (San Francisco, CA), Nov. 2001

[5] TCPDUMP Public Repository: `http://www.tcpdump.org`

[6] WinDump: tcpdump for Windows. Available at `http://www.winpcap.org/windump/`

[7] HP OpenView products: `http://www.hp.com`

[8] Strauss, J., Katabi, D. and Kaashoek, F. "A measurement study of available bandwidth estimation tools." In Internet Measurement Conference (2003)

[9] "TCPDUMP Subsampling Problem on the Deterlab Testbed", Soranun Jiwasurat, The Pennsylvania State University

[10] Will Leland, Murad Taqqu, Walter Willinger, and Daniel Wilson, "On the Self-Similar Nature of Ethernet Traffic (Extended Version)", IEEE/ACM Transactions on Networking, Vol. 2, No. 1, pp. 1-15, February 1994

[11] V. Paxson and S. Floyd, "Wide-area Traffic: The Failure of Poisson Modeling," IEEE/ACM Transactions on Networking, pp.226-244, June 1995

[12] The Internet: On its International Origins and Collaborative Vision, Ronda Hauben, `http://www.ais.org/~jrh/acn/ACn12-2.a03.txt`

[13] Man-Computer Symbiosis, J. C. R. Licklider, IRE Transactions on Human Factors in Electronics, volume HFE-1, pages 4-11, March 1960

[14] Leonard Kleinrock, "the Birth of the Internet", `http://www.lk.cs.ucla.edu/personal_history.html`

[15] Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, Stephen Wolff (2003). "A Brief History of the Internet".

[16] Simple Network Management Protocol, by Cisco: `http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/snmp.htm`

[17] Cisco, Netflow services and applications, Available from `http://www.cisco.com`

[18] Amer, P. D. and Cassel, L. N. (1989). Management of sampled real-time network measurements. In Proc. 14th IEEE Conference on Local Computer Networks 6268. IEEE Press, New York.

[19] Micheel, J., Braun, H.-W. and Graham, I. (2001). Storage and bandwidth requirements for passive Internet header traces. In Proc. Work-

shop on Network-Related Data Management. Available at `www.wand.cs.waikato.ac.nz/pubs/6/pdf/nrdm2001.pdf`

[20] D. Papagiannaki, N. Taft, S. Bhattacharyya, P. Thiran, K. Salamatian, and C. Diot, "A pragmatic definition of elephants in internet backbone traffic," in Workshop 2002. `http://citeseer.ist.psu.edu/papagiannaki02pragmatic.html`

[21] Wolff, R. (1982). Poisson arrivals see time averages. Oper. Res. 30 223231.

[22] S. Waldbusser, Remote Network Monitoring Management Information Base, IETF RFC 2819, May 2000.

[23] Network Monitoring Tools: `www.slac.stanford.edu/xorg/nmtf`

[24] J. Apsidorf, K. C. Claffy, K. Thompson, and R. Wilder, "OC3MON: Flexible, affordable, high performance statistics collection", in Proc. of INET, June 1997

[25] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, and J. Rexford, "Netscope: Traffic enginnering for IP networks", IEEE Network, March/April 2000.

[26] N.G. Duffield. C. Lund, M. Thorup. "Learn more, sample less: Control of vomule and variance in network measurement", IEEE Transactions in Information Theory, vol. 51, pp. 1756-1775, 2005.

[27] N.G. Duffield. C. Lund, M. Thorup. "Properties and Predictions of Flow Statistics from Sampled Packet Streams", ACM SIGCOMM Internet Measurement Workshop 2002, Marseille, France, November 6-8, 2002

[28] "Characterizing Traffic Workload", available at : `http://www.caida.org/analysis/learn/trafficworkload/`

[29] Fraleigh, C., Moon, S., Lyles, B., Cotton, C., Khan, M., Moll, D., Rockell, R., Seely, T., Diot, C. Packet-level traffic measurements from the sprint IP backbone. IEEE Network (2003)

[30] Cisco NetFlow services and applications customer profiles. Available at `www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/profiles`

[31] InMon Corporation (2004). sFlow accuracy and billing. Available at `www.inmon.com/pdf/sFlowBilling.pdf`

[32] Sampling for Passive Internet Measurement: A Review, N.G. Duffield, Statistical Science,Vol. 19, No. 3, 472-498, 2004.

[33] Schervish, M. J. (1995). Theory of Statistics. Springer, New York.

[34] Jedwab, J., Phaal, P. and Pinna, B. (1992). Traffic estimation for the largest sources on a network, using packet sampling with limited storage. Technical Report 92-35, HewlettPackard Laboratories, Bristol. Available at `www.hpl.hp.com/techreports/92/HPL-92-35.html`

[35] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker. On the characteristics and origins of internet flow rates. In Proceedings of ACM Sigcomm, August 2002

[36] "The CoMo White Paper", Gianluca Iannaccone, Christophe Diot, Derek McAuley, Andrew Moore, Ian Pratt, Luigi Rizzo, Intel Research technical Report, Intel Research, Cambridge, UK

[37] The GEANT Network: `http://www.geant.net`

[38] PlanetLab Design Notes: `http://www.planet-lab.org/PDN`

[39] Estan, C., Keys, K., Moore, D., and Varghese, G. 2004. Building a better NetFlow. SIGCOMM Comput. Commun. Rev. 34, 4 (Aug. 2004), 245-256. DOI= http://doi.acm.org/10.1145/1030194.1015495

[40] Provisioning IP Backbone Networks Based on Measurements, Konstantina Papagiannaki, PhD Thesis, University College London

[41] Kyoungwon Suh, Yang Guo, Jim Kurose, Don Towsley, Locating network monitors: complexity, heuristics, and coverage, IEEE INFOCOM 2005, Miami, USA.

[42] Xianghui Liu, Jianping Yin, Zhiping Cai and Shaohe Lv, On the Placement of Active Monitor in IP Network,ICCNMC 2005, Zhangjiajie, China