Statistical Sampling for Non-Intrusive Measurements in IP Networks

Dipl.-Ing. Tanja Zseby

von der Fakultät IV – Elektrotechnik und Informatik
der Technischen Universität Berlin
zur Erlangung des akademischen Grades

Doktorin der Ingenieurwissenschaften
– Dr.-Ing. –

genehmigte Dissertation

Promotionsausschuss:
Vorsitzender: Prof. Dr. Hans-Ulrich Heiß
Berichter: Prof. Dr.-Ing. Adam Wolisz
Berichter: Prof. Dr. Burkhard Stiller

Tag der wissenschaftlichen Aussprache: 06.Dezember 2005

Berlin 2005

D 83

"We are drowning in information and starving for knowledge."

Rutherford D. Roger

Abstract

Network measurements are essential for network operation. They provide vital data for network maintenance, fault detection and network planning. They support service provisioning by accounting functions and the validation of transmission qualities. Network measurements are an important element for the detection of network attacks and form the basis for network research.

Nowadays, heavier data rates demand measurement functions that operate at higher speeds. The amount of data traffic carried by the Internet each day has increased dramatically within the last decades. A deceleration of this trend is not in sight.

Higher data rates increase not only the amount of data that can be measured but also the amount of measurement results that need to be processed and stored per time unit. Utilization of dedicated hardware increases operational costs, precluding a broad-scale deployment, whilst unintentional exhaustion of resources leads to uncontrolled data loss with unpredictable effects on measurement results. The only way to get reasonable measurement results out of limited resources is by smart data selection.

The subject of this work is the investigation of statistical sampling methods for non-intrusive measurements in IP networks. The focus is set to packet selection methods for the estimation of traffic characteristics. The key challenges are to assess and predict the estimation accuracy and to keep information loss at minimum when only a subset of data is available. In order to address these challenges different data selection methods are investigated by theoretical modeling and empirical tests. They are compared with respect to their technical applicability to IP measurements, achievable estimation accuracy and expected resource consumption. A special focus is set on the investigation of stratified data selection methods. For these, classical statistics promise an accuracy gain without additional costs, if intelligent grouping of data can be applied by utilization of a-priori information.

A large part of this work was funded by Cisco Systems with the project VEGAS (**V**olume **E**stimation for the **G**eneration of **A**ccounting Data with **S**ampling). Cooperation with Cisco Systems allowed the incorporation of practical aspects. A number of ideas were successfully contributed to standardization and a patent was filed for the modeling of a specific selection scheme.

Zusammenfassung

Messungen sind von essentieller Bedeutung für den Betrieb von Kommunikationsnetzen. Sie liefern notwendige Daten für die Wartung, Fehlerdiagnose und Netzwerkplanung und bilden die Basis für die Netzwerkforschung. Sie unterstützen die Bereitstellung von Netzwerkdiensten durch Abrechnungsfunktionen und die Überprüfung von Übertragungsqualitäten und sind wichtiger Bestandteil bei der Erkennung von Netzwerkattacken.

Steigende Datenraten erfordern heutzutage Messfunktionen die bei höheren Geschwindigkeiten arbeiten. Die gewaltigen Datenmengen, die von einer steigenden Anzahl von Nutzern täglich über das Internet übertragen werden, erhöhen nicht nur die Menge der zu messenden Daten sondern auch die Menge der Messergebnisse. Eine zeitnahe Nachbearbeitung, Analyse und Speicherung aller Messdaten ist bei hohen Datenraten nicht mehr möglich. Die erheblichen operativen Kosten verhindern einen ausgedehnten Einsatz hardware-basierter Speziallösungen. Die einzige Lösung um mit begrenzten Ressourcen aussagekräftige Messergebnisse zu erzielen ist eine geschickte Selektion von Daten.

Ziel dieser Arbeit ist die Erforschung von Stichprobenverfahren für den Einsatz bei passiven Messungen in IP Netzen. Der Schwerpunkt wird auf den Einsatz von Paketselektionsverfahren zur Schätzung von Verkehrscharakteristiken gesetzt. Kern-herausforderungen sind hierbei die Beurteilung und Vorhersage von Schätzgenauigkeiten und die Minimierung des Informationsverlustes wenn nur Teilmengen von Daten verfügbar sind.

Um diese Herausforderungen anzugehen, werden verschiedene Datenselektionsverfahren mit Hilfe theoretischer Modellbildung und praktischer Experimente untersucht. Die Verfahren werden anhand technischer Anwendbarkeit, erzielbarer Schätzgenauigkeit und erwartetem Ressourcenverbrauch verglichen. Ein Schwerpunkt wird auf die Untersuchung von geschichteten Selektionsverfahren gesetzt. Hier verspricht die klassische Statistik einen Genauigkeitsgewinn ohne Zusatzkosten, falls eine intelligente Gruppierung der Daten durch Verwendung von a-priori Informationen möglich ist.

Ein großer Teil dieser Arbeit wurde von Cisco Systems über das Projekt VEGAS (Volume Estimation for the Generation of Accounting Data with Sampling) finanziert. Die Kooperation mit Cisco Systems ermöglichte einen praktischen Bezug der Arbeit. Einige Ideen konnten erfolgreich in die Standardisierung eingebracht werden und es wurde ein Patent für die Modellierung eines speziellen Selektionsverfahrens angemeldet.

Acknowledgment

First of all, I would like to thank my advisor Prof. Dr.-Ing. Adam Wolisz for his guidance and support. He taught me what kind of questions I needed to ask to improve my scientific methodology and always encouraged me in my work. I am also very grateful to Prof. Dr. Burkhard Stiller for reviewing this work as a second advisor.

With regard to practical aspects I am exceptionally obliged to Cisco Systems for funding parts of this work. The VEGAS project provided me the chance to combine my research interest with a practical project and gave me the opportunity to discuss results directly with Cisco engineers. Special thanks go to Benoit Claise and Andrew Johnson for all their questions and comments.

Many thanks go to the whole METEOR team, for many insightful discussions and for coping with several times where this work distracted me from leading this group with the quality they deserve. Special thanks go to Dr. Mikhail Smirnov and Prof. Dr. Georg Carle for their guidance and to Sebastian Zander, Thomas Hirsch, Guido Pohl and Carsten Schmoll for inspiring discussions and valuable comments.

Furthermore, I would like to express my gratitude to the National Laboratory for Applied Network Research (NLANR) MOAT, the Waikato Network Research Group (WAND) and all other groups that provide network traffic data for the research community. Without such publicly available measurement data, network research would probably never have reached its current state.

Last but not least I would like to thank my family. Special thanks go to my parents, who taught me the value and joy that resides in a good education and always supported and encouraged me. My most heartfelt gratitude goes to Matt for his enduring love and support. He also earns my deepest respect for learning how to fly an aircraft in less time than I needed for finalizing this work.

Disclaimer

Cisco IOS©, Cisco Systems© and Cisco NetFlow© are registered trademarks of Cisco Systems©. All other trademarks mentioned in this document are the property of their respective owners.

Contents

List of Figures

List of Tables

1 Introduction

The Internet has become a highly dynamic heterogeneous mixture of various network devices and transmission technologies. It interconnects hundreds of millions users all over the world and transports huge amounts of traffic from a wide variety of applications. It seems that the only unwavering constant in this fast growing and continuously changing environment is the Internet Protocol (IP), which dominates and will most likely continue to dominate the Internet. The physical technology below is a highly dynamic mixture of heterogeneous network devices and link technologies, ranging from low capacity wireless access and DSL lines to high speed fibers and broadband satellite connections. A similar diversity can be observed above the IP layer. New protocols and applications evolve and permanently change the traffic profiles observed in the Internet.

The traffic in this network is a complex superposition of different data flows, sequences of IP packets that originate from various applications and multiple sources. Flows competing for the available bandwidth are multiplexed and shaped by network nodes and altered by link characteristics. Packets are discarded, delayed or reordered. They can be fragmented, encapsulated and modified, so that the characteristics of a flow changes significantly on the way from source to destination.

Measuring was essential from the beginning. Simple tools like *ping* and *traceroute* helped to ensure proper operation of the Internet in its early days and are still widely used today. Measurement functions were needed for maintenance and planning. Now that the Internet has become a platform for commercial services, more sophisticated measurement functions are required as basis for accounting and quality auditing. Furthermore, network security applications demand continuous network surveillance and fine grained traffic inspection.

Another important field for network measurements, besides network operation, is networking research. As for nearly all other research disciplines, measurement provides the basis for scientific investigation. The influence of new technologies and new protocols can only be investigated, assessed and improved with the help of measurement results that provide insight into effects on network, applications and service quality. Whereas the importance of measurements is well recognized in most classical disciplines like physics, medicine and biology, its relevance for Internet research has long been undervalued. In an experiment performed in 2001, researchers from other disciplines evaluated networking research in [CSTB01]. The participants were shocked how few network researchers know about their subject of research and how limited their abilities are to measure the Internet. One of their three recommendations to network researchers was to improve their abilities to measure and to collect data about their subject of research.

Nowadays a wide variety of measurement tools and measurement platforms exists, adjusted and fine-tune towards many different measurement tasks and objectives. Standardization bodies like ITU-T and IETF have initiated several groups that deal with the manifold aspects

in network measurements. And of course there are innumerable publications about measurement results taken for various reasons.

Nevertheless, we are still very far from being able to provide a real in depth picture of the Internet and the traffic within. Measurement functions need to operate at higher speeds in order to cope with increasing data rates. Higher data rates also elevate the amount of result data and with this, the resource consumption for processing, storage and result data transmission. This work addresses this challenge by the investigation of data selection techniques.

1.1 Motivation

Measurements methods are called ***non-intrusive*** or passive if they are based only on existing traffic in the network and do not inject any test traffic. They provide an elegant way for investigating the quality and quantity of existing data flows without burdening the network with additional load. Non-intrusive measurements form the basis for many applications with increasing importance in future IP networks like usage-based accounting, the validation of quality guarantees specified in service level agreements (SLAs) and detection of network attacks.

The main challenge for the deployment of non-intrusive measurements is the increase of data rates. Measurement functions often cannot keep up with capturing, processing and storage if data arrives at high rates. The required resources for measurement operation and the transport of result data grow immense. The use of dedicated hardware does not really solve the problem. Measurement functions are usually only supplementary functions. Therefore, measurement costs should be limited to a small fraction of the costs of providing the network service itself. Expensive hardware solutions increase operational costs, precluding a deployment at broad scale. Furthermore, even if capturing of all data was possible, the enormous amount of measurement result data would still require immense resources for processing, storage and transport. Since the amount of observed traffic can change extremely over time, the amount of measurement result data can also vary. Growing measurement demands like fine granular classification of data or the need for new metrics further aggravates the problem.

In this work the challenge is addressed by an investigation of ***data selection techniques***. The main hypothesis is that using only a subset of the data is sufficient to serve the measurement needs of many applications.

1.2 Problem Statement

The goal of measurements is to determine characteristics of a population. The ***population*** is the set of all existing data elements. For network measurements usually the network traffic forms the population. The selection of only some elements from the population can cause a loss of information. The key challenge is to keep the loss of valuable information at minimum

when applying data selection. What can be considered as useful information in a specific scenario most likely differs with regard to scenario conditions (e.g., target application, measurement method, characteristic of interest, etc.) and has to be investigated. Furthermore a measurement process usually consists of multiple processing steps. The available amount and aggregation level of data may vary at different points in the sequence of operations. Deployment of selection methods at different points in the measurement process may have different effects.

Therefore, one dimension that needs to be considered for the assessment of schemes is the available ***traffic information***. It defines what is known about the network traffic (the population) before and after the selection process. One example for information that may be available before the sampling, are link rates or expected traffic profiles (e.g., only VoIP traffic expected). An example for information that may be available after the selection process has completed is the size of all sampled elements. The amount and type of available information is usually determined by the measurement process (which data is collected, stored and postprocessed, which functions are applied, etc.) and the position of the selection process within the measurement process (e.g., where is which information accessible).

If the remaining information in the selected subset of data is not sufficient to determine the exact value of the characteristic of interest, the characteristic has to be estimated from the available data in the subset. Due to the lack of information an estimation can always cause estimation errors. Therefore a major criterion to assess the quality of a data selection scheme is the assessment of size and distribution of potential estimation errors, expressed by the estimation ***accuracy***.

A further important dimension are the measurement ***costs*** that are required to collect the subset of data. Costs here represent the required resource consumption for storage, processing and transport of data. The cost reduction that can be achieved by a specific data selection scheme can be expressed by the fraction of elements from the whole population that need to be processed to calculate an estimate. Figure 1-1 shows the three relevant dimensions for the assessment of data selection schemes: costs, accuracy and available traffic information.

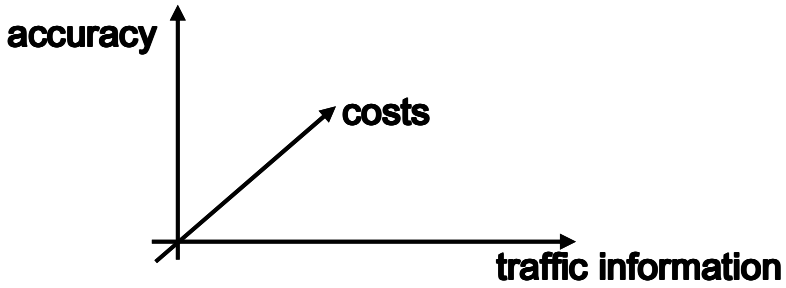


Figure 1-1: Dimensions for Assessment of Data Selection Schemes

A major part of this work is the investigation of interdependencies between those dimensions for selected scenarios. From this one can derive rules for selection of appropriate schemes and parameter settings.

There are different possibilities to select data from the population. Different random or deterministic selection methods can be applied. Schemes may require different information from the network, have different costs and achieve different accuracies. The definition of a ***taxonomy*** helps to categorize the wide variety of potential schemes and to apply criteria for scheme comparison.

If the precise determination of an exact value is substituted by an estimation, it is essential to provide information about the expected extent of estimation errors. This leads to one of the main challenges. The ***prediction and control of estimation accuracy*** is important for providers to assess possible revenue loss or gain and to inform customers about potential inaccuracies. A reasonable assessment of the estimation accuracy is a prerequisite for customers to tolerate data selection techniques. One can differentiate three different sub-problems with this. First of all ***accuracy calculation*** requires to develop statistical models to compute the expected accuracy at all. The models express whether and how the estimation accuracy depends on scheme parameters and characteristics of the population. If the accuracy depends on population characteristics, it can get a tough challenge to find generic accuracy assessment models that are valid for different populations. Conflicting aggregation levels between the characteristic of interest and the point where data selection is deployed (e.g., information loss by aggregation, information gain by classification, etc.) further complicates the problem.

The second sub-problem is ***accuracy prediction***. Accuracy prediction explains how the estimation accuracy can be assessed if only information is taken into account that is available before and after the selection process.

The third sub-problem is ***accuracy control***. The acceptance of some applications would profit enormously if a stable accuracy could be guaranteed. If the accuracy depends on population characteristics and those characteristics vary over time (as it is usually the case for network traffic), it is quite likely that also the accuracy varies. One possibility to provide accuracy control is the adaptation of scheme parameters. This is based on an accuracy prediction from

previous results and the knowledge about dependencies between accuracy and scheme parameters.

In the same way one has to look at the ***prediction and control of resource consumption***. A predictable, stable or controllable resource consumption is desired for resource planning and to prevent exhaustion of resources.

Investigation of accuracy and resource consumption for selected scenarios is addressed in this work. Typically one would expect that low costs and high accuracy are contradicting goals that cannot be achieved at once. In the same way it is probably difficult to aim at constant accuracy and constant resource consumption at the same time.

1.3 Contribution of this Work

In this work I investigate data selection methods for ***non-intrusive*** measurements in IP networks. I consider single data packets as basic elements and consider the set of data packets that are observed at a specific observation point in the network within a given measurement interval as population. Selection schemes that consider packets as basic elements are called ***packet selection*** schemes.

First I suggest a measurement framework and a ***taxonomy*** for packet selection schemes, based on related work and current standardization efforts. Then I investigate packet selection methods for two applications, for which nowadays it is most crucial to keep up with high network speeds and reduce measurement efforts: ***usage-based accounting*** and ***SLA validation*** (i.e., the validation whether quality guarantees given in SLAs are fulfilled).

Based on an evaluation of existing work in this area and on the developed taxonomy, missing but promising basic schemes are identified. Those schemes are investigated with regard to the three dimensions described above by theoretical modeling and experiments with real traffic traces.

For the first investigations traffic information is taken as given by the measurement process. Models are developed that show the dependencies between achievable accuracy, characteristics of the population (if relevant) and required resource consumption. Those models would provide a precise calculation of accuracy and resource consumption if all information about the population was available. Since probably not all population characteristics are known if only a subset of data is captured, it is then investigated to what extend an accuracy prediction is possible if only information is taken into account that is available before or after the data selection process in reality.

Although accuracy prediction provides valuable input for accuracy control, accuracy control methods themselves are not addressed in this work. The development of suitable accuracy control techniques requires further work on traffic prediction methods, which is itself a broad field of research.

In a second step, I try to tackle the main challenge of data selection, the tradeoff between resource consumption (costs) and accuracy, by a deeper exploitation of traffic information. Based on the hypothesis that the incorporation of additional information helps to increase the estimation accuracy, I look into a technique called ***stratified sampling***. Stratified sampling is a powerful data selection method for accuracy improvement in classical statistics and performs the data selection in two steps. The accuracy gain is achieved by an intelligent grouping of elements before the selection process in accordance to a known or easy to obtain characteristic of the population (stratification variable) that has some correlation with the characteristic of interest. Up to now only few have looked into stratified sampling for network measurements. Existing work lacks theoretical modeling and only investigated arrival-time or packet-count as stratification variables. I investigate the use of stratified methods for the selected target scenarios, identify further stratification variables and analyze the achievable accuracy gain by theoretical modeling and empirical investigations.

1.4 Document Structure

The remainder of this document is structured as follows:

In Chapter 2 an overview of IP networks and IP measurement methods is given. Measurement challenges are identified and state of art in network measurements is reviewed. Existing measurement methods are explained and a measurement reference model is defined under consideration of current standardization efforts.

In Chapter 3 the need for the deployment of sampling methods is explained and a taxonomy for packet selection schemes is introduced. The taxonomy explains the relevant parameters for packet selection schemes and provides a categorization of schemes in accordance to input parameters and selection method. Then a detailed analysis of the relevant scientific publications in this area is given. The research plan is presented, which explains the scope of this work and the applied methodology.

In Chapter 4 the investigations of packet selection schemes for usage-based accounting is documented. First criteria for the usage of selection schemes in accounting scenarios are defined. Based on this appropriate basic schemes are selected, for which the highest benefit is expected. For the selected scheme theoretical models for the calculation of the expected accuracy are developed based on hypothetical assumptions, which later are validated.

In Chapter 5 the models developed in chapter 5 are validated by empirical investigations. It is analyzed whether real traffic traces fulfill initial assumptions that were needed to derive the models. It is empirically investigated how traffic characteristics influence the estimation accuracy. Furthermore, it is analyzed what effects occur in cases where initial assumptions for the theoretical modeling do not hold.

In Chapter 6 the theoretical modeling for the use of data selection techniques for SLA validation is developed. Requirements and criteria for the deployment of selection methods

for SLA validation scenarios are derived. Based on this appropriate methods for an in-depth investigation are selected and mathematical models for the calculation of the estimation accuracy are presented.

In Chapter 7 the empirical results for the SLA validation are presented. Real traffic traces are investigated with respect to assumptions made for the model building and for border cases where assumptions do not hold. Furthermore, experiments are performed for different stratification strategies.

In Chapter 8 the results are summarized and an outlook on potential future work items is given.

2 IP Measurements

This section provides an overview of IP measurements. It first introduces IP networks, shows the need for network measurements and gives an overview of related work in this area. For terms that are not explicitly defined here, the terminology introduced in [Tann03] is used.

2.1 IP Networks

This section describes how IP networks work. It gives and overview about network elements and protocols and shows how IP packets and flows form the traffic in a network. The term IP networks is used as a general term for all networks that run the IP protocol. The Internet is considered as a concatenation of multiple IP networks. That means a single provider network is considered as an IP network but not as “the Internet”.

2.1.1 Network Elements and Protocols

IP Networks consist of network nodes (routers and switches), which are connected by fixed network links or wireless technologies. ***Internet service providers (ISP)*** offer customers access to networks and network services like data transmission. Multiple hosts form local area networks, using technologies like Ethernet and wireless LAN. ***Routers*** connect networks (Figure 2-1). They forward and multiplex IP packets received on their network interfaces. An ***autonomous system*** (AS) is a network or a collection of networks that implements a common routing policy [RFC1930]. Routers within such a network that have no connection to another AS are called ***core routers***. They are optimized to operate at high speeds. Due to the high load on such links, core routers are fully dedicated to fast packet forwarding. CPU power should not be shared with additional functions, like QoS functions or advanced measurements. ***Border routers*** establish connections to other ISPs at the edge of the network. They implement routing policies for peering relationships (i.e., agreements between providers). Those routers have a key position for accounting between providers and inter-domain SLA provisioning. If such features are needed, border routers or additional measurement devices located close to the border routers should provide measurement functions. ***Edge routers*** connect customer networks and individual users to the ISP. They terminate a large number of links and need to enforce traffic limits or perform QoS functions like the marking of packets for prioritization. They need to provide measurement function for accounting and SLA validation. ***Access routers*** are the routers at customer’s premises that connect the customer network to the edge router. ***End systems*** are usually connected by Local Area Network (LAN) technologies, run various applications and provide the entry point for users to the network.

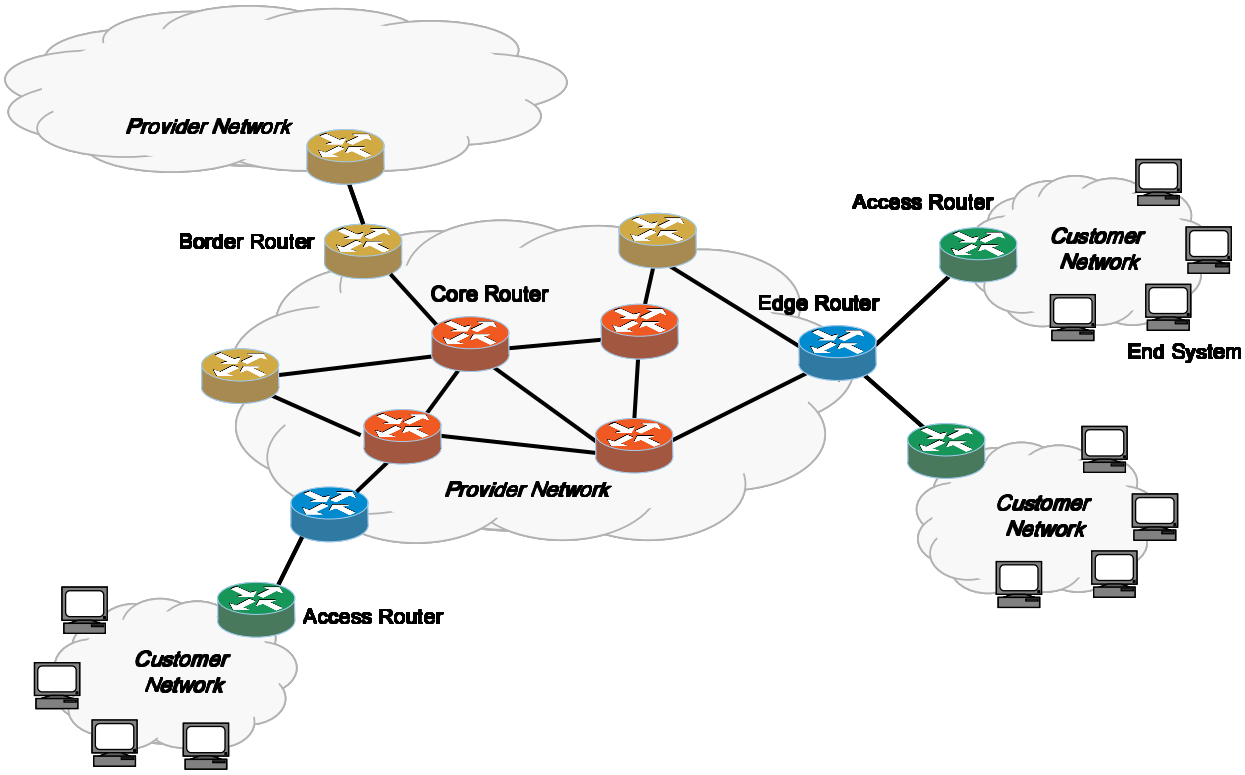


Figure 2-1: IP Network Structure

Routers maintain routing tables that contain information how other networks can be reached. Those tables are either statically configured or established by ***routing protocols*** which run between routers and calculate the best paths to forward the packet. If a packet arrives on a network interface, the router examines the IP destination address in the IP packet header and forwards the packet in accordance to the information in the routing table to the appropriate output interface. This processing delays the packet at each hop it has to traverse on its path to the destination. Since many packets arrive at multiple input interfaces, which may need to be forwarded to the same output interface, the router usually needs to queue arriving packets before they can be sent out from the output interface. Since buffer memory is limited, routers have to discard packets if the network gets congested. This queuing and reaction to congestion situations in routers is the main cause for packet delay, jitter and loss in the network.

Basis for the communication in the Internet is the ***Internet Protocol (IP)*** [RFC791]. IP provides a "best effort" connectionless packet delivery service. Each packet travels independently. Packets can take different paths in the network although source and destination are the same. The service is unreliable because packets may be lost, delayed, duplicated and re-ordered on the way to the destination. IP does not provide any detection or re-transmission of lost packets.

The current IP version in the Internet is IPv4. Due to the immense increase of hosts and devices on the Internet an address shortage is expected. Mainly due to this reason, but also to incorporate further features, a new version of the Internet Protocol, ***IPv6*** [RFC2460], has been

defined and is currently deployed. IPv6 has much larger address ranges (128bits instead of 32 bits). IPv6 introduces the concept of dynamic packet headers which increases the challenges for measurement operations (e.g., classification). Furthermore the coexistence of IPv4 and IPv6 networks requires tunneling of IPv6 packets over IPv4 packets. Such encapsulation further complicates in-depth analysis of packet header and context.

The ***Internet Control Message Protocol (ICMP)*** [RFC792] became an integral part of IP at the very beginning. It provides error reporting and measurement support. The well known and widely used measurement tool *ping* uses this protocol to check availability of hosts on the network and round-trip delays between source and destination.

IP Multicast [RFC1112] provides a method to distribute packets to multiple receivers in a very efficient way. Instead of sending multiple copies of a packet to all receivers, the sender only sends one packet, which then is duplicated in routers only if necessary at junction points on the way to the receivers. This unburdens sender and network and allows an efficient communication with large groups. On the other hand more functionality at network nodes is required for duplicating packets and maintenance of packet distribution trees. Multicast requires new metrics and new measurement methods to support the operation of applications that use IP multicast. Nevertheless, multicast communication introduced some new problems. Network management and routing gets more complex. Provisioning of reliable transport, security features, accounting, and quality of service is much more difficult than for unicast communication. Due to the additional problems, IP multicast is currently not that widely deployed as one could expect from a technology that leads to such an efficiency improvement. To react to the problems, less complex solutions for specific scenarios (e.g., single source multicast) are currently investigated in research and standardization. Furthermore, application layer multicast is used for multipoint communication, but is lacking the high efficiency that can be achieved by multicast at IP layer.

IPsec allows the encryption [RFC2406] and authentication [RFC2402] of IP packets on the way from sender to receiver. It is a very important feature on the Internet in order to provide security and privacy. Since encryption and decryption is usually done at the communication end points, measurement functions in network nodes on the path are unable to look into some parts of the packet header and into the payload. Getting information about encrypted traffic by doing measurements contradicts the purpose of encryption and is therefore a difficult and often unsolvable challenge.

Below the IP layer multiple different technologies and link layer protocols are in use. Above IP, the ***User Datagram Protocol (UDP)*** [RFC768] and the ***Transmission Control Protocol (TCP)*** [RFC793] are the dominant protocols. Knowledge about content and functionality of those protocols, helps later to understand how network traffic can be structured (e.g., definition of flows, see 2.1.2). TCP allows a reliable and congestion-aware data transport. It establishes a connection between source and destination. The reception of TCP packets is acknowledged and packets are re-transmitted if needed. Furthermore, the transmission rate

adapts to network conditions. A reliable transport is essential for applications that need to ensure that all data is received (e.g., file transfer). Fairness and congestion awareness are important in networks where multiple connections compete for the same bandwidth. Since the rate adaptation and re-transmissions can lead to delays in packet transmission, timely delivery of data cannot be ensured with TCP, making it unsuitable for real-time applications like audio and video streaming.

UDP provides no control mechanisms and no reliable transport. Packets are simply sent to the destination and silently discarded if congestion occurs. Therefore UDP is not suitable for applications that require reliable transport. For real-time applications like audio or video transmission timely delivery is much more important. The loss of a few packets can be tolerated by those applications. Therefore UDP is much better suited in these cases. On top of TCP and UDP a wide variety of applications are present in the Internet.

Additional protocols came up that combine some of the features of both protocols. The ***Real-time Transport Protocol (RTP)*** [RFC3550] provides some lightweight control features (like sequence numbers) to improve real-time transmissions. The ***Stream Control Transmission Protocol (SCTP)*** [RFC2960] provides a new transport protocol with some TCP features and additional functions especially for telephony signaling. Furthermore, SCTP allows an operation where it is only partial reliable (SCTP-PR) [RFC3758].

2.1.2 Network Traffic

IP packets that originate from different applications and sources form the network traffic. They consist of a packet header and the packet payload. The combination of packet header and payload is also called packet content [ZsMD05]. The IP packet header contains, among other fields, the source and destination IP addresses and the protocol in use. The IP header is followed by a UDP or TCP transport header that contains source and destination port numbers and other protocol specific fields (Figure 2-2). Network elements can alter packets. Routers change some packet header fields (e.g., time to live (TTL) and checksum). They can fragment, encapsulate, prioritize or encrypt packets. Routers can apply functions to change the addresses (e.g., for Network Address Translation (NAT)), compress headers and many more.

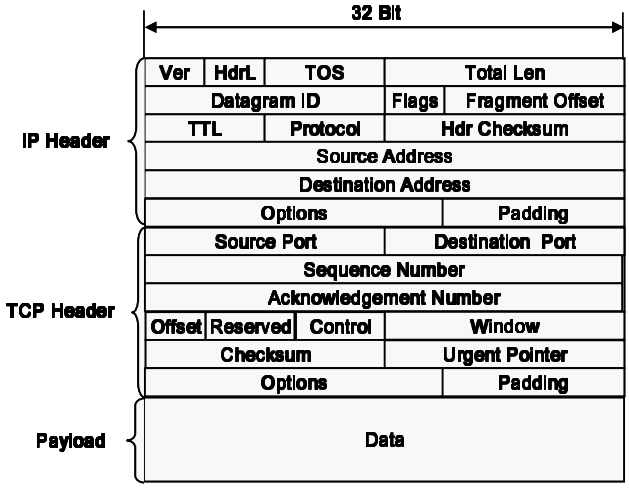


Figure 2-2: TCP/IP Header

Packet attributes are properties of the packet like the packet size or IP type. One can differentiate between attributes that are only related to the packet and those which are related to one or more observation points (i.e., a location in the network where packets are observed, see definition in A). Examples for packets attributes are all packet header fields, like IP type, packet size, or the packet payload. Examples for attributes related to observation points are the arrival time of the packet at one observation point, the delay that a packet experiences between two observation points or the path that a packet takes through the network.

Flows are a set of one or more packets with common properties [RFC3917]. The flow definition defines how coarse or fine granular the classification into different flows is done. Flow definitions can range from the set of all packets on a link to considering each individual packet as a separate flow. The packet attributes that are used to differentiate flows are often referred to as flow keys (e.g., in Cisco NetFlow [RFC3954]).

Throughout this document the generic flow definition given in [RFC3917] is used. Nevertheless, other flow definitions exist. Further flow definitions can be found for instance in [ClPB95] or [RyCB01]. [RFC2722] introduces a bi-directional flows definition.

Flows that belong to a specific application and are distinguished by the 5-tuple consisting of source address, destination address, source port, destination port, protocol, are often referred to as micro flows. A flow that includes all packets (observed at an observation point) for a specific traffic class is also called traffic aggregate.

Flow characteristics are the traffic characteristics for a specific flow. Flows can have much more attributes than packets, because they usually consist of multiple packets. Therefore flow characteristics can be aggregated packet attributes or parameters of the distributions of packet attributes can be considered as flow characteristics.

Flow characteristics are in general random variables. In many networks the distribution of the number of packets per flow or the number of bytes per flow are heavy-tailed. That means, most flows consist only of a small number of packets and only few flows have a large number

of packets. The few large flows contribute to the majority to the overall traffic volume [FaPe99], [DuLT01]. This observation on the flow size distributions in Internet traffic is also referred to as "Quasi-Zipf-Law" [KuXW04] or as "elephant and mice phenomenon". The large flows are referred to as elephant flows or ***heavy hitters***. Nevertheless, such observations depend on the flow definition in use and can change with regard to the profile of future applications.

The traffic in Internet backbones is a superposition of many different flows, originating from a variety of applications from users all over the world. The amount of flows and the flow characteristics can be very dynamic. Traffic from different sources is mixed and shaped by routers. Packets are queued, fragmented and discarded to fit them to the capacity of the underlying transmission technology. So a wide range of factors influence the traffic characteristics, including the location of senders and receivers, user behavior, applications and protocols in use, routing decisions and forwarding algorithms, the underlying transmission technologies on the way from sender to receiver and of course the presence of concurrent flows. Most of these factors are highly dynamic and many of them remain unknown. Floyd and Paxson show in [FlPa01] how the extremely dynamic nature of the Internet introduces difficulties to measurement and simulation. So the traffic mix observed at one point in the network is not only highly dynamic but also quite difficult to predict.

Network traffic was often modeled as Poisson or Markov arrival processes [FrMe95]. Nevertheless, there is now evidence that network traffic has ***self-similarity*** (e.g., [LeTW94], [PaFl95]). Self-similarity means that traffic patterns (e.g., for packet count or transmitted bytes) at different time scales resemble themselves. The Hurst parameter provides a metric to measure the self-similarity of network traffic. Some implications of this are that burstiness can be observed at different time scales and traffic does not necessarily get smoother if multiple sources are aggregated. Furthermore, self-similar traffic exhibits long range dependencies (LRD), i.e., the autocorrelation function drops only slowly and may never reach zero. It is still an active field of research to find appropriate ways to model self-similar traffic.

2.1.3 Quality of Service

The development of methods to prioritize specific flows is a very active field of research. New resource intensive applications arise, that can exhaust even high-speed network connections. High quality video transfer, grid computing and gaming require high bandwidth and transmission qualities. For the communication between machines, data reception is not restricted to human perception limits. Furthermore, overprovisioning of resources does not work if communication demands increase suddenly (e.g., in emergency situations).

There exist various concepts to provide a prioritized service to some packets. An overview of techniques to deploy quality of service can be found in [Tann03]. In the Integrated Services (IntServ) approach (e.g., [RFC1633], [RFC2205]), reservations are made in routers on the path for packets that belong to a specific flow (e.g., defined by source, destination address,

port numbers, etc.). The IntServ approach has strong support for multicast. A disadvantage of the IntServ concept is that it requires configuration in the network and state keeping in routers.

An alternative is the Differentiated Services (DiffServ) approach (e.g., [RFC2475], [RFC2638], [RFC3260]). In this approach packets are marked before they enter the provider network and with this assigned to a priority class. For this the type of service (TOS) field in the IPv4 header is used. IPv6 provides a special field for indicating the priority. The bit field used for indicating the priority of a service is called the DiffServ codepoint. Routers decide with regard to the DiffServ codepoint, how the packet is treated and assign them to different queues. Provider and customer specify in a service level agreement (SLA) how the priority classes are defined in terms of quality parameters like delay, loss, jitter.

Nevertheless, although several methods exist for QoS provisioning, such techniques are currently not widely deployed. One reason is that providers usually try to provide more bandwidth than needed (overprovisioning), so that all data flows get a sufficient transmission quality. Some researchers also argue that there is not enough demand from applications, that QoS approaches are not yet technically mature enough and suitable accounting strategies are lacking [MaBP03] or that the QoS research just had a bad timing [CrHM03]. Problems with overprovisioning is that it requires a good network planning, may be costly and does not work for sudden and unexpected high demands. Prioritization of specific services and users may be for instance required in emergency communication (e.g., [RFC3487], [RFC3689], [RFC3690]). [HeBh03] discusses gaming as a potential QoS demanding application. In [ShTe03] it is proposed to use QoS as protection against denial of service attacks.

2.2 The Need for Network Measurements

Network measurements are a vital part of Internet operation and networking research. They are essential for ***maintenance and planning***, traffic engineering and profiling. With the evolution of distinguished transport services and the ongoing commercialization of the Internet it became necessary to support such services by functions for ***accounting*** and ***SLA validation***. Measurements also provide insight into traffic profiles which reflect user behavior and service usage. Such information can form the basis for planning and assessment of new services.

A further field with growing importance is ***network security***. An increasing number of various network attacks (e.g., DoS, worm propagation) endanger service provisioning and can cause severe damage to the Internet. There are different attack detection methods based on network measurements. One approach is the recognition of attack patterns (signature detection). Such methods require knowledge about the expected attack traffic and therefore only work for known attacks. Another method is anomaly detection. With anomaly detection one tries to detect deviations from the expected behavior. Anomaly detection can also detect unknown attacks (zero-day events) but relies on a good knowledge and prediction of what is considered

as normal behavior. The decision whether traffic patterns indicate an attack or not can become a tough challenge, because attackers adapt their strategies to detection systems and try to conceal attack traffic by using so called stealth attacks.

Furthermore, network measurements are required for ***network research***. Measurement techniques are essential for the investigation of new technologies and new protocols. They are needed to assess the transmission quality achieved with new technologies and their effect on existing protocols. They are also required to evaluate how which new protocols and applications are used and how they change traffic profiles.

2.2.1 Measurement Applications

The IPFIX requirements document [RFC3917] identifies target applications for passive measurements and defines their demands on collection and export of measurement result data. In accordance to this, the following key applications are defined:

- Fault management and network maintenance
- Usage-based accounting
- QoS measurements and SLA validation
- Traffic profiling and traffic engineering
- Attack and Intrusion detection

An introduction to the applications is given in [RFC3917]. Measurement requirements for selected applications can be found for instance in [RFC3917] and [OsHe02]. Table 2-1 summarizes what information and measurement features are needed for those applications (M- Mandatory, R-Recommended, O-Optional, - - not of interest).

Required Information		Fault Management and Maintenance	Usage-based Accounting	SLA Validation/ QoS Measurements	Traffic Profiling and Engineering	Attack/Intrusion Detection
Link Information	Connectivity	M	-	-	R	-
	Packet count per link	M	O	-	M	M
	Byte count per link	O	O	-	R	R
Flow Information	Packet count per flow	-	R	O	R	M
	Byte count per flow	-	M	-	R	R
Packet Information	Header	-	M[1]	M[1]	M	M[1]
	Content	-	O	O	O	R
	Arrival Time	-	M	M	O	O
Multipoint	Clock Sync	R	M	M	O	O
	Multipoint correlation	R	O	M[2]	R	R
	Inter-domain Exchange	O	O	R	O	R

Table 2-1: Measurement Requirements for Key Applications

Fault management and network maintenance requires link information about connectivity between network nodes and link counters. Flow and packet information are useful for diagnosis of problems but not necessarily needed to detect problems. It is useful to measure at multiple points and provide synchronized clocks to monitor when an incident was observed at a specific observation point. Basis for most network management application is the Simple Network Management Protocol (SNMP). Management Information Bases (MIBs) in network nodes provide information (e.g., link counters). This information can be requested from management applications by using the SNMP protocol.

Usage-based accounting requires information about the transmitted data volume per flow. Packet header information is required to classify packets into flows. Flows are defined in accordance to the tariff model and typically distinguish traffic from different source networks or packets that belong to the same service class. Monitoring the correct arrival time and

[1] For classification

[2] For some metrics

providing synchronized clocks is required to cope with time-based tariffs that depend on usage duration or time of the day.

Traffic profiling and traffic engineering require packet counts per link in order to check the network load at specific nodes. This information can be used for planning and load sharing decisions. Information from packet headers is needed to check what protocols are in use. If further packet information is available one can further analyze the traffic up to the identification of applications in use.

QoS measurements and SLA validation encompasses the measurement of different quality metrics like loss, delay and jitter (see 2.2.2). Packet header information is required for classification, because SLAs are typically valid for a specific traffic class and/or customer. Arrival times are needed to capture delay and loss. One-way measurements require multiple observation points and clock synchronization.

Attack and Intrusion detection requires packet counts per link and flow to detect whether there is a sudden increase of packets to a specific location. A deeper look into the packet payload is desirable in order to check for specific attack patterns (e.g., the ratio of TCP-SYN to TCP-FIN packets). A correlation of data from multiple observation point is useful. Data exchange between providers can help to track an attacker faster and to isolate the source of the attack.

2.2.2 Network QoS Metrics

Network QoS metrics describe the quality of the data transmission. Different applications demand different quality criteria to operate in a way that customers are satisfied. There are many publications about the definition and use of QoS metrics at various layers. In this work the standardized metric definitions defined by the IETF IP Performance Metrics group [IPPM] are used. The IPPM group focuses on the standardization of objective metrics that provide an unbiased quantitative measure of the network performance. Subjective metrics like perceived audio or video quality, mean opinion scores, etc. are out of scope of this work. A framework for defining metrics is provided in [RFC2330]. The group has defined a set of standard metrics for the assessment of the transmission quality in IP networks:

- ***Connectivity***: [RFC2678] defines metrics to determine whether hosts (IP addresses) can reach each other.
- ***One-way delay***: One-way delay (OWD) is defined in [RFC2679] as the time difference from the time where the source sends the first bit of the packet until the time where the destination received the last bit of the packet. If a packet is duplicated in the network and multiple copies of the packet arrive at the destination, the copy that arrives first at the destination is used to calculate the delay. The IPPM group recommends the use of GPS synchronization at the measurement points. For non-

intrusive measurements the time difference between the first and the second observation point have to be considered instead of the times at source and destination.

- ***One-way loss***: One-way packet loss is defined in [RFC2680]. The packet loss is 0 if a packet that was sent by the source is received at the destination within a pre-defined time interval and 1 if the packet is not received at the destination within this time. If the received packet is corrupted it will be counted as lost. The measurement methodology has to distinguish between a packet loss and a very large (but finite) delay by setting appropriate thresholds.
- ***Round-trip delay***: Round-trip delay is defined in [RFC2681] as the time difference from the time where the source sends the first bit of a packet until the time where the source received the last bit of a packet that was immediately sent back by the destination after receiving the packet from the source.
- ***Delay variation***: IP packet delay variation (IPDV) is described in [RFC3393]. The IPDV is defined for a selected pair of packets that belong to the same stream. It measures the difference between the one-way-delay of the selected packets. The term ***"Jitter"*** is not used by the IPPM community, because it is used in different contexts and can have further different meanings.
- ***Loss patterns***: In [RFC3357] two metrics are defined that are derived from the loss metric described in [RFC2680]. The ***loss distance*** is defined as the difference in sequence numbers of two successively lost packets which may or may not be separated by successfully received packets. The ***loss period*** is a sequence of subsequently lost packets. If a packet is successfully received and after this a packet is lost, a new loss period starts. These two metrics are used to describe loss patterns.
- ***Packet reordering***: The definition of a re-ordering metric is currently discussed within the IPPM working group. The goal is to provide metrics to show if and to what extend packets are re-ordered on the way from source to destination. The Internet draft [MoCR04] combines different ideas for the definition of such metrics.

The IPPM working group cooperates with other groups in the IETF (e.g., Benchmarking Methodology Working Group [BMWG], Remote Monitoring Management Information Base [RMON], Traffic Engineering Working Group [TEWG]) and with other standardization bodies and forums, such as T1A1.3, ITU-T SG 12 and SG 13, in order to provide consistent metric definitions.

Delay is caused by different physical properties and operations in an IP network. One can decompose the delay between two observation points in the network into the following components (e.g., [ChMZ04]).

- ***Propagation delay***: Propagation delay reflects the physical characteristics of the network link. It depends on the physical medium in use and the distance that is

covered. The propagation delay usually does not vary in a way significant for the IP packet transmission.

- ***Transmission delay***: The transmission of the packet on the link incurs additional delay depending on link capacity and the packet size.
- ***Processing delay***: Processing delay reflects the time needed to process the packet at network nodes. It is composed of the time to evaluate the packet header and determine an appropriate output interface. The header checksum needs to be recalculated and the time to live field (TTL) is decreased. Nowadays high-speed routers complete this task in less than 40 µs [PaMF02].
- ***Queuing delay***: Queuing delay reflects the delay a packet experience due to queuing in the router. It depends on the load in the router and therefore is highly variable.

2.2.3 Service Level Agreements (SLAs)

In order to meet the needs for quality-demanding applications and to allow customers to assess the service they can expect, providers offer guarantees about the transmission quality in their networks in ***Service Level Agreements (SLAs)***. SLAs are contracts between network providers and customers that define what quality a customer can expect from the network. They usually contain guarantees for specific QoS parameters over given time intervals. The part of SLAs that contains the technical statements about QoS guarantees is also called service level specification (SLS).

In order to check whether a guaranteed transmission quality is really fulfilled, providers need to measure with which quality the customer traffic is transferred in the network. In some SLAs also the measurement techniques themselves are specified. In order to base SLAs on estimated QoS values, provider could specify confidence levels and error margins in SLAs.

2.3 Measurement Methods

This section describes measurement methods for the measurement of different metrics. The distinguishing between non-intrusive (passive) and intrusive (active) measurements is explained. A measurement reference model is derived under consideration of current standardization efforts. Furthermore, it is shown which additional requirements need to be considered for multipoint measurements.

2.3.1 Passive vs. Active Measurements

Active measurements inject test traffic into the network in order to measure network characteristics. Traffic patterns and execution times can be adapted to specific measurement objectives. Active measurements allow the performance of controllable experiments and are well suited for monitoring the network situation, fault detection etc. Nevertheless, they are not appropriate for traffic-related applications like accounting, SLA validation and intrusion detection. Usage-based accounting unquestionably has to be based on the existing traffic in

the network. Intrusion detection needs to examine existing traffic for possible suspicious patterns. SLA validation is possible with active measurements to a certain degree, but there are several disadvantages.

First of all test traffic always induces additional load on network links and routers. This can cause further congestion problems in times when network load is high. The additional traffic interferes with the customer traffic and therefore can influence the transmission quality and measurement results. A second problem is the generation of appropriate test traffic. In order to obtain representative measurement results the artificial traffic must emulate the expected customer traffic, or at least work with patterns, that allow a suitable quality statement about the customer traffic. In most cases the generation of appropriate test traffic is not trivial. A third problem is the treatment of test traffic. For appropriate measurements it has to be ensured that the injected test traffic is treated equal to the real user traffic. This is especially important if independent third parties or customers themselves perform the measurements. If test traffic is not recognizable as such, providers may suspect an attack because the additional traffic cannot be matched to a customer and discard the packets. If test packets are clearly marked, providers may give them a preferred treatment in order to manipulate the statistics.

Passive measurements are based only on the observation of already existing traffic. They provide real information how current traffic is served in the observed network. This is exactly the information that is needed for accounting and SLA validation. Since no test traffic is generated, passive measurements can only be applied when the traffic of interest is already present in the network. This is the case for accounting and SLA validation, because both are only needed during the service usage, i.e., when the traffic of interest is present. Furthermore, measurement results for SLA validation are especially needed when quality reduction impends e.g., due to congestion. In such situations it is unwise to further increase the network load by sending test traffic. Therefore passive measurements are here considered as method of choice not only for accounting but also for SLA validation.

In short one can conclude that active measurements are well suited to measure the network state in a controllable way whereas passive measurements are more appropriate to provide information about the existing traffic in the network.

2.3.2 Measurement Reference Model

In this section a generalized measurement architecture is presented. Basic terminology and building blocks from IETF standardization is used. Different IETF groups (RTFM, IPFIX, PSAMP, AAA, AAAARCH) focus on different parts of the measurement infrastructure and sometimes propose slightly different architectural concepts. I try to combine those concepts into a consistent overall picture. Since this work considers only non-intrusive measurement methods, the architecture is focused on components for passive measurements. For active measurements, one can consider the receiver of the test traffic (either the sender itself or another probe), where measurement results are collected, as passive observation point and

apply a similar architecture. An approach for controlling active measurements can be found in [ShTK04]. Terminology, a framework for metrics and recommendations for generation of test traffic can be found in [RFC2330].

2.3.2.1 Measurement Architecture

Figure 2-3 shows the measurement reference architecture. The ***measurement process*** comprises all functions needed to perform the measurements (see section 2.3.2.2). The measurement process runs on an ***observation point***. The observation point defines the location in the network where the traffic is observed. The observation point can be for instance a network card in a measurement probe or an interface on the router [RFC3917].

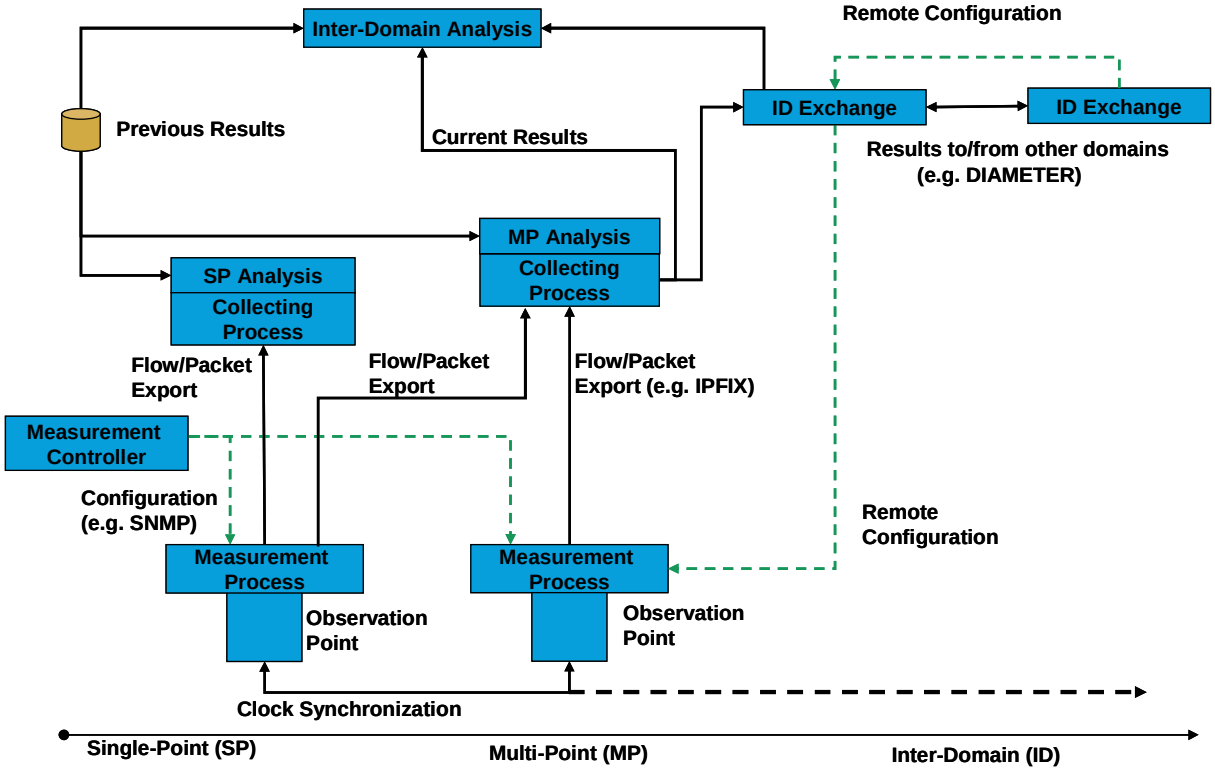


Figure 2-3: Measurement Reference Architecture

Measurement results are transferred by a packet or flow ***export protocol***. The Real-time Traffic Flow Measurement Architecture [RFC2722] and the remote monitoring management information base [RMON] use the Simple Network Management Protocol (SNMP) for data export. Some vendors have defined proprietary protocols for exporting measurement data. Examples are Cisco NetFlow or InMon sFlow. Now the IETF standardizes the IP Flow Information Export (IPFIX) protocol as future standard for the export of flow information [Clai05]. Cisco systems are preparing to use the IPFIX protocol on their routers. It is likely that other vendors will also use the IPFIX standard for data export in future The IETF PSAMP group has recently decided to also use the IPFIX protocol for the export of per packet information. With this, IPFIX provides the future standard for the export of packet and flow information.

Packet records are data structures that contain information about packets. This can be a copy of the whole packet itself, selected parts of the packet (e.g., the header), packet properties (like arrival time or packet size) or some information derived from the packet (e.g., a packet ID calculated from the packet content). ***Flow records*** contain information about flows. Examples for flow information are the number of packets in the flow, the flow duration or the mean packet size in the flow.

Flow and packet records are reported to one or more ***collecting processes*** within the network. Collected data is analyzed in accordance to the measurement objective, i.e., the required metrics are calculated. Usually collecting process and measurement process run on different devices. Nevertheless, collecting processes and analysis functions can be co-located with measurement processes. That means that data collection and the calculation of metrics may for instance take place at a dedicated router (see 2.4.4.1).

The calculation of metrics can incorporate data from one (***single-point*** measurements) or multiple observation points (***multi-point*** measurements) or even from other domains (***inter-domain*** measurements). It can also incorporate results from past measurements.

The measurement process is configured by a measurement ***controller*** that distributes the measurement tasks to different measurement processes. The configuration can be done for instance by SNMP. If results are needed from two or more observation points to calculate the metric, in most cases ***clock synchronization*** among the involved observation points is required. Clock synchronization methods are described in 2.4.4.3.

For inter-domain measurements it is required to exchange result data across multiple administrative domains. For the data transmission across domains more stringent security requirements have to be fulfilled. It is also possible to allow a remote configuration of measurement processes in foreign domains. The authentication, authorization and accounting (AAA) architecture provides a secure data transfer between domains by using the DIAMETER protocol. It is possible to use this architecture for the inter-domain exchange of result data and measurement configuration [RFC3334]. Such approaches are discussed in section 2.4.3.

2.3.2.2 Measurement Process

The measurement process consists of multiple functions that are needed to transform the observed packet stream into packet or flow records. Figure 2-4 shows the components of a measurement process in accordance to the understanding of the IETF IPFIX and PSAMP group with some modifications in order to combine both views. [RFC3917] specifies observation point, flows, exporting and collecting process. The document also defines a metering process that consists of packet header capturing, timestamping, classifying, sampling and maintaining flow records (see Figure 1 in [RFC3917] and Figure 5 in [SaBC05]). The PSAMP group defines a measurement process as a composition of a selection and reporting process. In order to consider the PSAMP and IPFIX architecture together I

consider packet capturing with a configurable snapsize (number of bytes that should be captured). The picture shows the measurement and export of packet information (left side) and flow information (right side). Core functions are always part of the measurement process. Optional functions can be placed in the processing sequence for different operations like post processing or data selection.

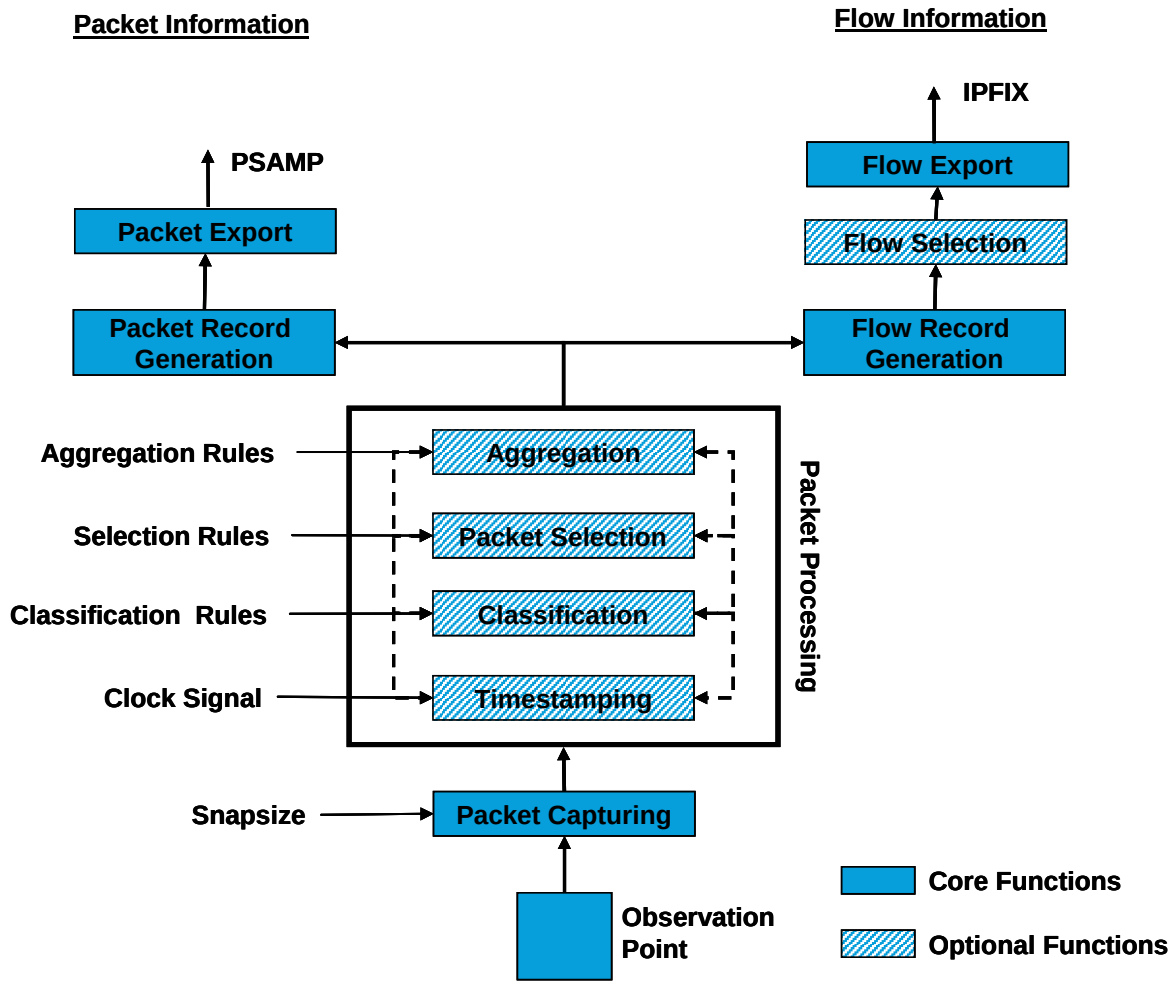


Figure 2-4: Measurement Process

Packet capturing records the arriving packets at the observation point. The snapsize defines how many bytes are captured per packet. With the snapsize the packet capturing can be restricted to the packet header or to parts of the packet payload. After the packet capturing different packet processing functions can be applied. All packet processing steps are optional and can be applied in any sequence and number of instances.

Timestamping adds the arrival time to the packet information. It is required if time-related metrics are calculated like delay or jitter or if the time of the day is relevant for the application (e.g., accounting for time-of-day dependent tariffs). Timestamping should be done as early as possible in order to get the best possible accuracy for the arrival time. Variable delays in the processing of packets (e.g. due to varying CPU load) reduce the timestamping accuracy. The longer packets are processed before the timestamp is applied, the higher are those effects from

hardware and operating system. Timestamping requires a clock signal which should come from a synchronized reference source (see section 2.4.4.3).

Packet selection methods select a subset of the captured packets. Different methods are distinguished in [ZsMD05]. ***Filtering*** is a deterministic packet selection based on packet content only. All other selection methods (i.e., random selection based on packet content or any selection not based on content) are called ***sampling*** [ZsMD05]. It is possible to concatenated different selection methods (filtering or sampling functions) in arbitrary order. For instance one could apply a filter before sampling or two different sampling methods subsequently.

Classification groups all incoming packets into classes in accordance to pre-defined classification rules. It assigns a flow ID to each packet that defines to which flow the packet belongs. No packet is removed during the process.

Classification and Filtering are different operations. The output of a classification process contains all incoming packets and additionally the information to which class the packet belongs to. The output of a filtering process is the set of only those packets that fulfill the filter condition. Classification adds information about all elements of the population whereas filtering removes elements. Figure 2-5 illustrates the difference.

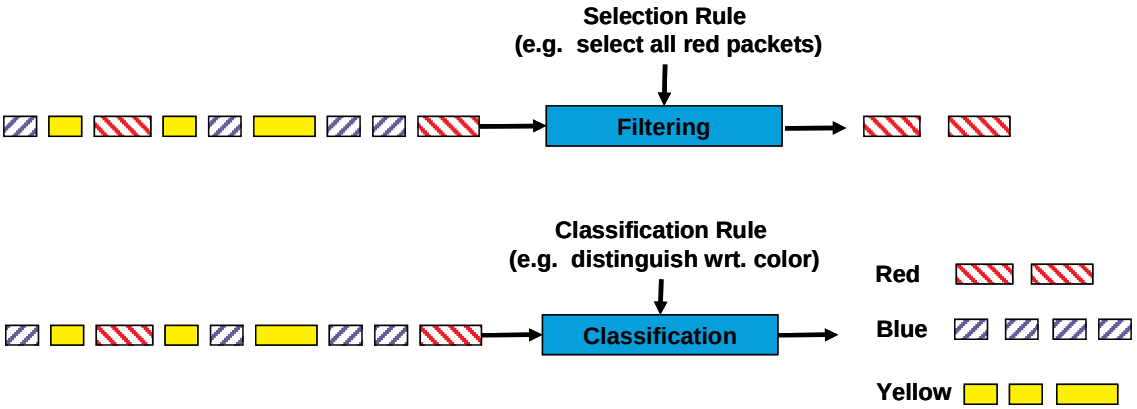


Figure 2-5: Classification vs. Filtering

Classification rules define how the different classes are distinguished. An example is the differentiation of flows with regard to the source address. After the classification process one knows how many packets originated from source A, B, C, etc. Classification can be realized with multiple non-overlapping filtering operations where each packet of the population is selected by exactly one filter.

Aggregation describes the combination of data into a composite [Duff04]. ***Aggregation*** methods are used to comprise per packet information (e.g. generation of a packet digest) or to combine information from multiple packets. Examples are the summing up of all packet sizes from all observed packets or the calculation of parameters of packet size distribution.

A formal description of aggregation methods is given in [Duff04]. Aggregation is a technique for data reduction, but since all elements are processed it is not a selection method. The

individual elements from which the aggregate is derived are discarded. With this the amount of data is reduced (e.g., sum of packet sizes instead of all packet sizes from all packets).

Aggregation can be seen as an opposite function to classification. Classification adds information about the class to which the element belongs and aggregation removes information, e.g., about individual packet sizes. So timestamping and classification provide additional information about the packets whereas packet selection and aggregation methods remove information.

Often classification and aggregation functions are applied together. For instance, to find out the transmitted data volume per flow one first groups packets into flows by classification. Then one sums up all packet sizes from the flow by an aggregation function.

Flow selection methods select a subset of all flows for flow export. It is used to reduce memory consumption for the flow cache and resources for transfer of flow records.

In order to unburden subsequent processes, selection functions should be applied as early as possible in the sequence of operations. If packet selection is done before classification only the selected packets need to be classified. If flow selection is done before flow export, only the selected flows need to be transmitted.

2.4 Measurement Challenges and State of Art

In recent years many papers have been published that present measurement approaches for different purposes. General challenges for network measurements are presented for instance in [MuCl01], [Claf02], [Claf03] and [CSTB01]. Floyd and Paxson show in [FlPa01] how the extremely dynamic nature of the Internet introduces difficulties to measurement and simulation. In a recent paper Vern Paxson summarizes strategies for the performance of sound Internet measurements [Paxs04]. Measurement research mainly deals with the development of new metrics and measurement techniques (e.g., [Jaco97], [Paxs99], [DuGr00], [EsKM04], [MoCR04]), developed to cope with speed, mobility and security requirements in current and future networks. Challenges related to speed are the development of inexpensive measurement hardware and fast packet classification algorithms. Further topics are the control of distributed measurement infrastructures (e.g., [AdMa00], [PaAM00], [RFC3763]), inter-domain aspects and related problems, like measurement synchronization and clock synchronization.

Since measurements are basis for research, many networking research groups have also developed own measurement tools or adapted existing tools for their needs. It would go way beyond the scope of this work to mention all those tools. The coordinating action on Monitoring and Measurement (MOME), which coordinates measurement activities in Europe, has identified nearly 400 different measurement tools that are coming from various research groups. An overview of the tools can be found at [TOOLS]. The MOME project provides a database that allows searching for suitable tools for different purposes and evaluates the

interoperability of the investigated tools [MOME]. An overview of existing measurement groups and measurement tools can also be found in [Osle02]. A comparison of measurement infrastructures is available at [INFRA].

In this chapter current measurement research challenges are presented and an introduction to existing work in this area is provided. The focus is set to non-intrusive measurements, because they are subject of this work.

2.4.1 Keeping Up with High Packet Rates

The main challenge for passive measurements is to keep up with high packet rates and an increasing number of flows. Required resources for storage, post-processing and transport of results increase with the amount of data that is captured per packet. In some cases additional information (e.g., arrival times) need to be stored. This can lead to an overwhelming amount of result data that can even grow larger than the data transmitted on the network itself, e.g., if the whole packet content is transmitted with additional information like timestamps.

One approach to deal with increasing packet rates is the development of ***specialized hardware***. In [ClDG00] it is pointed out that design goals for standard PCs and network interface cards contradict measurement needs. One example for this is that interface cards can arbitrarily discard packets if rates are too high, whereas for precise measurements it is important to capture all incoming packets. Furthermore, an early and accurate timestamping of received packets is important for measurements and not available in standard PCs. The paper concludes that only specialized hardware can fulfill measurement demands.

Luca Deri describes in [Deri03a] lessons learned from trying to use standard hardware for measurements in Gigabit networks. He recommends assessing measurement system performance in packets/second and not in Mbit, because it is more difficult to measure many small packets than few large ones. He sees one main bottleneck at the file system performance that impedes the storage of data at Gbit speed. Furthermore, he criticizes the “divide et impera” concept, where traffic that should be measured is distributed among several probes, because it requires multiple probes and does not reduce the amount of result data. He rather suggests the use of traffic preprocessors like nProbe [Deri03a], which optimize packet capturing and flow cache operations. As a further possibility he mentions the use of sampling methods.

Endace Systems [ENDACE] provides hardware cards, the DAG boards [DAG], that are specialized for packet capturing and are widely used within the research community. The DAG boards are based on a configurable and programmable structure and include an on-board filtering of data before it is passed to further processing. They allow highly accurate timestamping of packet arrivals with special support functions for the use of GPS-based clock synchronization. If GPS is available the cards provide timestamping with an accuracy of ±250ns [McGr02]. DAG boards are available for a variety of transmission technologies with a consistent architecture for speeds up to OC192/STM64 and 10G Ethernet. The fact that many

scientists use the same measurement hardware also increases consistency and comparability of measurement results.

The European project SCAMPI ([SCAMPI], [CoMN04]) develops a high-speed measurement system and looks into the use of network processors for classification and filtering operations (e.g., [NgCB04]). [FrDL01] describes a special OC-3 and OC-48 monitoring system deployed at Sprint. The problem with solutions based on hardware are the high costs, which preclude a broad deployment of such systems.

Another approach to cope with increasing packet rates is the ***improvement of packet processing algorithms*** (e.g., storage and classification). Since packet processing is needed for a variety of network functions (e.g., routing, QoS provisioning, etc.), the optimization of packet classification and filtering techniques has become a broad field of research. A well known packet filter implemented in Unix systems is the Berkley Packet Filter (BPF). BPF is used by the tool *tcpdump* [JaLM01]. New packet filtering and classification techniques are described for instance in [Srin01], [ErMS01], [WaSV01], [Woo00] and [BoBC04]. The new techniques mainly focus on the improvement of search algorithms to find the appropriate entry within classification rules and the intelligent organization of data storage after classification.

[FeMu00] describes the tradeoffs between time and space requirements for packet classification. A method presented in [CoSV04] tries to reduces the required number of evaluated filter expressions to operate at higher speeds. An overview of classification and filtering algorithms can be found in [Schm01].

Furthermore, due higher packet rates and fine grained flow definition, a higher number of flows can be observed (e.g., [FaPe99]). More efficient methods are needed to store per flow information and to reduce the number of flow cache operations needed (e.g., lookup). [IaDG01] describes a technique to reduce storage requirements by efficiently organizing per packet and per flow information to capture and store flow information on 10GB-Ethernet and OC-192 links. Space-Code Bloom Filter (SCBF) introduced in [KuXW04] are described in section 3.6.

The third approach to cope with increasing packet rates is to apply ***packet selection*** techniques. This is focus of this work and further described in chapter 3. The increasing number of packets and flows also causes increasing number of result data (e.g., flow records, which report per flow information). With this the transport requirements increase. Some approaches propose to deploy ***flow selection*** methods to reduce memory and transport requirements (e.g., [DuLT01], [EsVa03]). Existing approaches for packet and flow selection are described in chapter 3.6.

2.4.2 New Protocols and Security

Further measurement challenges originate from new protocols. The flexible header structure of IPv6 packets and the tunneling of packets (e.g., to transfer IPv6 packets over IPv4 networks, or multicast packets over unicast networks) requires more complex packet processing. ***IPv6 measurements*** are for instance addressed in [6QM].

The use of ***IP multicast*** increases measurement complexity, because traffic is send from and to multiple nodes. Packets are duplicated within the network in order to save resources on the way from senders to receivers. There are several properties of IP multicast that complicate measurement tasks in multicast environments [SaAl00]. Multicast routing establishes multicast trees instead of paths. Instead of measuring the characteristics of a single path between source and destination one now has to deal with a multicast tree and need to consider branches to single receivers separately. Due to the dynamic group membership the tree structure can change frequently. Receivers can leave or join the multicast group at any time. The number and distribution of nodes is highly dynamic. When measuring connectivity (e.g., with multicast ping), there can be an unexpected large number of reports that overload the sender. A further problem is the anonymity of group members. Senders do not know which receivers are listening. In contrast to the unicast case, senders cannot identify whether a link is broken, because they do not know in advance from how many and which nodes an answer is expected. Furthermore, due to the lack of a reverse path receivers may not be able to respond to measurement requests. An introduction to the multicast measurement problematic and an overview of multicast measurement tools can be found in [SaAl00].

A really tough challenge is the ***analysis of encrypted traffic*** (e.g., IPsec). Since original packet content and parts of the header cannot be read by intermediate systems, one measurement systems on the way can only extract very few information. Since concealing certain information is the purpose of encryption, one can only derive few assumptions about applications in use and other information from packet arrival patterns and the few unencrypted parts of the packets.

Packet traces contain a lot of information that can reveal user behavior. Due to privacy concerns nowadays an ***anonymization*** of packet traces is often required before they can be used for research. Approaches for anonymization can be found for instance in [Peuh01] and [XuFA01].

2.4.3 Mobility and Inter-Domain Aspects

The trend towards more mobility in the Internet has also effects on required measurement methods. One problem with mobile devices and wireless technologies are the ***scarce resources***. The available resources in mobile and wireless environments are usually more limited than in fixed networks. Devices have to be small and energy-saving. Measurement functions on mobile devices must be extremely resource-efficient. Furthermore, wireless technologies usually offer a poorer transmission quality and/or bandwidth than fixed

networks. Active measurements or the transfer of measurement results can very fast exceed available capacities in wireless networks.

A broad field in research deals with the measurement of the transport characteristics of new wireless technologies, quality problems during handover and the investigation of differences to fixed networks if standard protocols are used over wireless technologies. Nevertheless, those measurements can be usually performed with standard methods or slightly adapted measurement tools.

A further problem are roaming scenarios. The mobile customer has a contract with its home provider and connects to a foreign network. If the home provider requires measurement results e.g., for accounting or SLA validation purposes, measurement functions need to be provided by the foreign provider in the network where the customer connects. Providers often use tariffing as a separation criterion from competitors. They use different tariff systems which require different measurement results. In such cases the foreign provider needs to generate exactly the type of results that the home provider needs e.g., for a detailed invoice or for specific quality parameters. It would be ideal if providers allowed neighbor providers to request specific results and then configured their measurement functions in accordance to this demands. An approach to allow ***secure remote configuration*** by using AAA components is proposed in [RFC3334].

Also other applications would profit from ***inter-domain exchange*** of measurement data and remote configuration. One example is measurement-based detection of network attacks. An exchange of measurement results and the ability to get specific measurement results from neighbor providers can help to assess the dimension of an attack and may even allow the identification of the source(s) of malicious traffic. Nevertheless, nowadays cooperation between ISPs and especially the sharing of data is only performed at minimum level. Objections to sharing information are caused by concerns that competing ISPs may get too much insight into network structure, network operation and information about customers. Providers are concerned that competitors exploit this information to raise their own position in the market.

Nevertheless, this position may change. Further challenges like inter-domain service provisioning and attack detection need to be faced in future. Technical needs have already triggered the exchange of data in the past (e.g., exchange of routing information by the Border Gateway Protocol BGP). Increasing security threats like denial of service attacks or worm propagation will raise the need for a closer cooperation between providers. Sharing information is the key for good defense strategy (see e.g., [Yurc04]) and providers seem to become more open to share data. The large network operator Sprint for instance recently made an attempt to start a new IETF working group on inter-domain measurements.

2.4.4 Multipoint Measurements

Several measurement tasks require the collection and correlation of data from multiple observation points. Passive multipoint measurements have been addressed for instance in [GrDM98], [DuGr00] and [ZsZC01]. For such measurements it is required to provide a coordinated control of the measurement functions and a synchronization of the measurement processes at different observation points. The following additional issues need to be considered for multipoint measurements

- ***Placement of post-processing functions:*** In multipoint measurements data from different observation points are collected and further processed together. There are different options to locate post-processing within the network.
- ***Transfer of result data:*** Different architectures need to be distinguished regarding the transfer of result data to the post-processing functions.
- ***Clock synchronization:*** If the metric of interest is time-dependent (like delay) the clocks at the measurement points must be synchronized to ensure a correct calculation.
- ***Packet event correlation:*** For some metrics it must be possible to associate measurement reports generated from the same packet at different observation points to each other.
- ***Sampling Synchronization:*** If sampling is deployed it must be ensured, that the same packets are selected at all observation points.

Different challenges occur from these requirements. Possible solutions are described in the following sections. Sampling synchronization approaches are discussed in section 3.5 after the introduction of sampling methods.

2.4.4.1 Placement of Post-Processing Functions

Post-processing functions (e.g., the calculation of metrics) can be located at a separate dedicated machine (Figure 2-6, a) or co-located with one of the measurement processes (Figure 2-6, b).

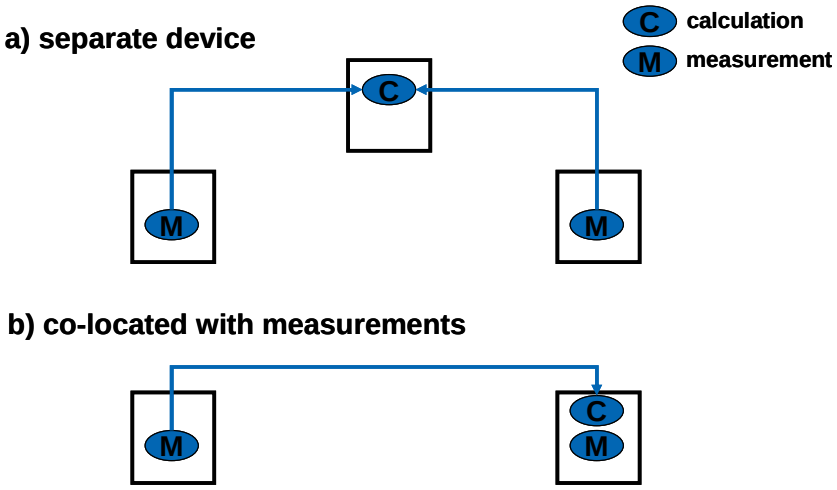


Figure 2-6: Location of the Post-Processing Functions

If the calculation is done on a separate device, the measurement results from both observation points need to be transferred to the calculation process. If a calculation process is co-located with one of the measurement points only the results from one observation point need to be transferred over the network. This saves network resources.

On the other hand the calculation process requires processing resources. Therefore this solution should only be used if sufficient resources for the calculation process are available at that observation point. Especially in heterogeneous measurement scenarios with different meter types this can be a suitable solution.

2.4.4.2 Transfer of Result Data

The transfer of the result data to the post processing functions can be done via the production network (Figure 2-7, a) or by using a separate network for the measurement data transfer (Figure 2-7, b) (i.e., different interfaces and links).

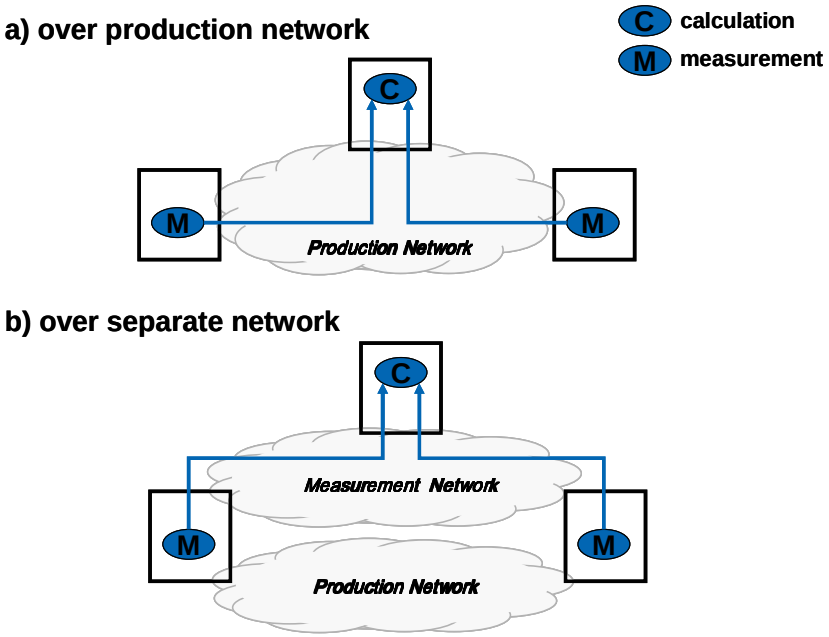


Figure 2-7: Measurement Result Transfer

Transferring measurement results over the production network incurs similar problems than the sending of test traffic in active measurements. The additional load in the network can bias the measurements and worsen the situation in times of congestion. Nevertheless, there are more options for the result data transfer than for the sending of test traffic. Measurement data for instance can be transferred in times where no measures are taken to prevent biasing the measurement results. Result data can be routed on different paths than the measured flows. If methods for service differentiation are deployed, result data can be transferred with a lower priority. Dependent on the location of the post processing functions, result data can also be transferred in the opposite direction than the measured flows.

2.4.4.3 Clock Synchronization

Clock synchronization is required for measurements that need to correlate arrival times at different observation points. An example is one-way delay, where the arrival times at different points in the network are needed to calculate the delay between network nodes.

A few atomic clocks, which are positioned e.g., at standardization institutes, provide a precise time reference. In order to synchronize clocks all over the world this precise time signal is distributed to interested receivers. There are different methods to distribute time information to network nodes.

- The ***Network Time Protocol*** (NTP) [RFC1305] uses the network for distribution of time information. NTP distributes packets with time information from a primary time server (stratum1 server), which is usually synchronized by GPS, to interested receivers. NTP also provides further features like methods for server discovery, a rough assessment of delay between client and server or the authentication of servers using symmetric key cryptography.

- The ***Global Positioning System*** (GPS) is a satellite navigation system controlled by the U. S. Department of Defense (DOD). The positioning is based on the latencies of signals from different satellites. Therefore GPS signals contain very precise time information, which can be received all over the world.
- ***Radio signals*** (e.g., DCF77) provide sources for time information that are used for private radio clocks, for many public clocks but also for scientific experiments. Availability of those signals is limited to the range of the radio signal but such signals exist in many countries all over the world and are synchronized with each other.

Each solution has advantages and disadvantages. NTP is very cost efficient. It needs no specific hardware except for one reference system. But NTP packets may experience different delays on the way from the NTP server to its destination. This packet delay leads to inaccuracies in the clock synchronization especially in large networks. In order to minimize this problem NTP includes methods to estimate the delay between client and server from RTT measurement. Recently, in 2004, a new IETF group was formed to standardize new versions of NTP. The group will first standardize NTPv4 which incorporates features already used today and than address further issues like IPv6 support for NTP and an enhanced security model. Further information about NTP can be found at [NTP].

GPS provides a more accurate solution. GPS-based solutions allow a precision with a maximum deviation of 100 nanoseconds. Nevertheless, for high precision measurements one has to take into account that there is a loss in accuracy on the way from the GPS receiver to the clock tick in the kernel. One disadvantage of GPS is that it is still more expensive compared to other solutions. Furthermore, although GPS signals can be received everywhere on earth, GPS receivers require a direct “intervisibility” with the satellites. This is a problem, because many server rooms or rooms that host main network nodes lack a window or are located at the basement. Measurement functions either operate directly on such nodes or measurement boxes are deployed close to those nodes. Therefore a direct visibility of satellites is usually not given. Cables that connect the GPS receiver to an antenna on the roof lead to additional inaccuracies.

Another option is to use specific radio signals that provide time information. For example the DCF77 signal is broadcasted at 77.5 kHz from a station near Frankfurt/Main in Germany. The signal can be received only in central Europe, but other countries also issue radio time signals that are synchronized with the DCF77 signal. That means as long as the network nodes have access to a radio clock signal, their clocks can be synchronized. Nevertheless, a proper reception of the signal cannot always be ensured, because atmospherics (e.g., from thunderstorms) or other radiation (e.g., magnetic or electric emission from devices close to the receiver) may interfere with the long-wave radio signal (see e.g., [DCFa]). More information about DCF77 can be found at [DCFb]. A comparison between DCF77 and GPS is available at [DCFa].

2.4.4.4 Packet Event Correlation

Some metrics require the correlation of packet arrivals at different observation points. For this one needs a unique identifier per packet in order to recognize that the same packet was observed at different points in the network.

With a copy of the packet itself it would be possible to identify it at different points in the network. In most cases it would be sufficient to use only some parts of the packet for identification. Nevertheless, transferring the whole packets or large parts of the packet only to recognize whether the same packet was observed at multiple observation points introduces way too much overhead. Therefore more efficient ways to identify packets are needed.

The datagram identification field in the IP packet header alone is not sufficient as unique packet identifier. It is only unique for a specific combination of header fields and limited to 16 bit. Considerations for the usage of the datagram identification for packet event correlation are given in [ZsZC01].

Another possibility is to generate a packet digest over packet header fields and part of the content and use this digest as packet identifier (ID). The packet ID generation should be done in a way that

- The resulting ID is as small as possible.
- The probability for collisions (getting the same packet ID for different packets) is as low as possible.
- The ID generation is fast.

Furthermore, it would be advantageous (but not required) to use an operation that always leads to an ID with the same fixed length. This would ease the handling, transmission and the estimation of the overhead caused by measurement result transport.

For the generation of a packet ID different fields of the packet and different methods can be used. In order to generate a unique packet ID that can be recognized at multiple observation points only the parts in the IP packet that do not change on the way to the receiver (immutable during transport) can be taken into account. Fields that are mutable but predictable could also be used for the packet ID generation. Furthermore, it is advantageous to consider fields that are highly variable between different successive packets. Whether fields with low variability (e.g., version field) should be considered in the ID calculation depends on the implementation. They should be included as long as there is no significant performance decrease. Different functions for packet ID generation were evaluated and compared for instance in [DuGr00], [ZsZC01], [NiMD04] and [NiMT04].

2.4.5 Active Measurements

Active measurements are usually performed to assess the status of the network, e.g., for network maintenance and fault detection. The main problem for active measurements is the generation of appropriate test traffic. This includes challenges like the definition of a send

schedule and packet types and the question whether test traffic should be recognizable as such.

It is important to keep the amount of test traffic at minimum in order to prevent overloading the network or influencing existing traffic. Metrics for active measurements are standardized in the [IPPM] group in the IETF. The documents of the IPPM group also contain proposals for send schedules for the measurement of specific metrics. Some researchers request the incorporation of active measurements into the network beyond ICMP. In [LuMc02] such a new protocol for integrated active measurements is introduced. Since the scope of this work is on passive measurements the challenges of active measurements are not further addressed.

2.5 Target Scenarios

In this work I concentrate on two target applications, which nowadays have become extremely relevant for service provisioning:

- ***Usage-based Accounting:*** Accounting based on the usage of network resources
- ***SLA Validation:*** Validation of the transmission quality

Both applications are described below.

2.5.1 Usage-based Accounting

There are many approaches to charge for the transmission of data. Network providers often use their tariff models as a way to distinguish themselves from competitors. Flat rate tariffs charge a constant fee for a given time period (usually per month). Usage-based tariffs take the amount of used resources into account. Most common schemes are based on transmitted volume or time duration. Tariffing strategies can provide incentives to users to use the network at off peak times (e.g., at night). With this the network load can be distributed more equally over time and congestions can be reduced. Using tariffing for the control of the network usage is a broad field of research. Ideas to control resource demand and supply can be based on technical and economic models and range from tariffs that consider the time of the day up to exceptional approaches where packet forwarding is based on an auction model [MaVa94].

Accounting describes the collection of data about resource consumption that is required to apply usage-based tariffs [RFC3334]. In this work accounting is considered to be done at network layer. The metric of interest is the transmitted data volume (in bytes) per flow. Since packet sizes can vary it is usually not sufficient to base accounting only on the transmitted number of packets. As defined in [RFC3917] the flow definition is quite flexible and can range from very coarse definitions that put all packets on the link into the same class to very fine grained classifications where packets are differentiated with regard to the generating application. The flow definition in use highly depends on the tariff model of the network operator (e.g., billing for sent or received volume, billing per application flow or per DiffServ

traffic aggregate). So the classification granularity has a high impact on the complexity of the measurement of per flow volume.

Accounting requires a ***passive single-point measurement***. Observation points are usually located at the entry points of the provider network (Figure 2-8). For accounting of customer networks measurement functions need to be deployed at edge routers. For accounting of traffic from neighbor providers measurement functions operate at border routers. In some cases access routers at user premises are controlled by the provider. In this case one could also install accounting functions at the access router. Measurements can be done co-located with routing or other functionalities or as exclusive function in a dedicated measurement device (network probe).

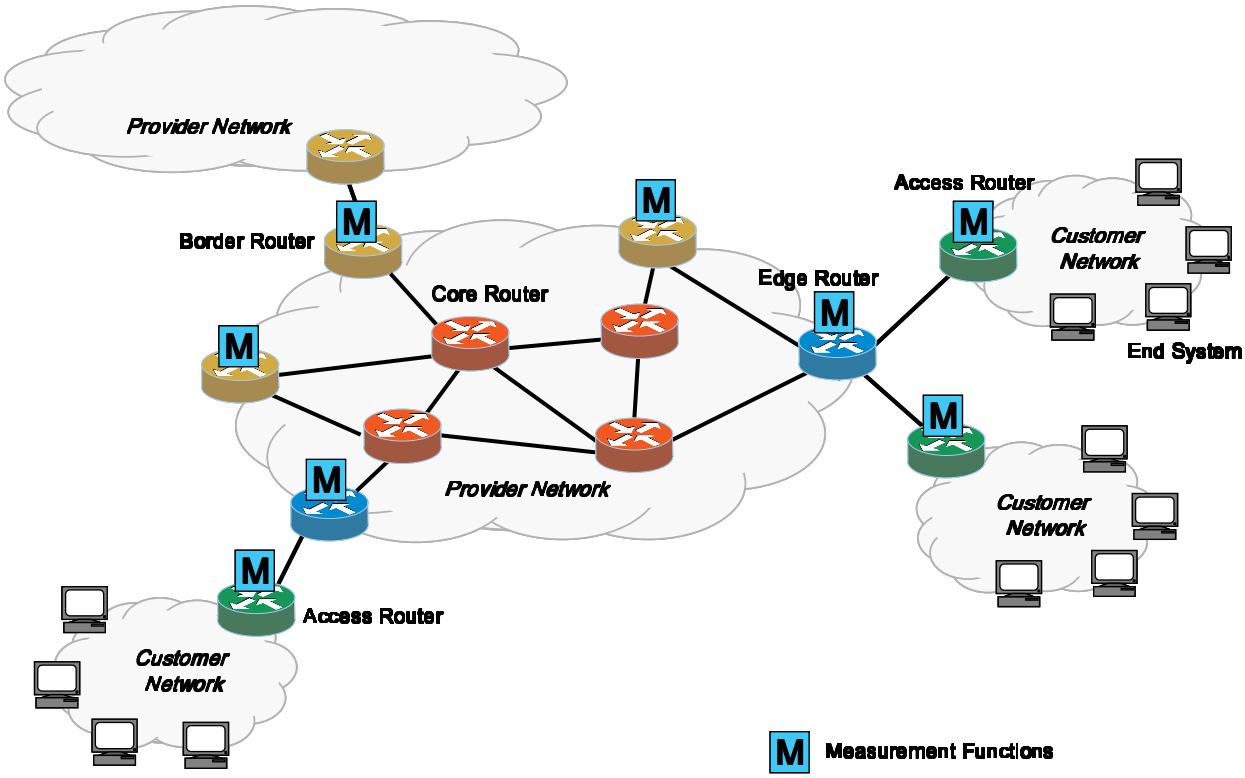


Figure 2-8: Potential Observation Points for Usage-based Accounting

In order to determine which packets belong to which flow, all packets that arrive at the observation point need to be classified. Classification rules are determined from the flow definition. Then the size field in each packet is evaluated and the packet sizes are summed up. This needs to be done separately for each flow that has to be measured. Byte counters per flow are stored in a flow cache and updated if a new packet for this flow is observed.

The amount and type of traffic that is observed depends on the type of customer that connects to the provider network. The customer can be one or more individual users, an enterprise network or a neighbor provider. Especially in the last two cases, the traffic at the measurement point can be immense. Packet classification and examination of the packet size becomes a problem if packet rates are too high. The maintenance of the flow cache becomes

difficult if the number of active flows gets too large (e.g., due to fine grained flow definitions).

Accounting is an auxiliary function for service provisioning. Therefore, it is desirable to reduce the resource demands (e.g., computational costs) and with this the monetary expenses for measurement functions.

Accounting provides the basis for billing. Billing is the generation of an invoice based on the accounting data. Therefore, customers may not tolerate estimates instead of exact measurements. Nevertheless, often measurement devices cannot keep up with high packet rates, e.g., if packets need to be classified for per flow accounting. Substituting the accurate measurement by an estimation may become attractive to customers if providers can reduce service provisioning prices. This is quite likely because provides can reduce prices due to the reduced accounting costs while maintaining their same revenue. An essential prerequisite to the use of sampling methods here is that the expected estimation accuracy can be predicted in advance so that customers and providers can assess potential monetary loss due to estimation errors. Therefore the sampling goal for accounting is to reduce the number of packets that need to be examined regarding their size, while maintaining a given accuracy for the computation/estimation of the overall volume of a flow. The metric of interest here is the volume (in bytes) per flow.

2.5.2 SLA Validation

For a variety of application providers nowadays guarantee specific quality levels for the data transmission. The service level agreement (SLA) is a contract between customer and network provider that defines which quality is guaranteed. Measurements are needed by customers and providers to check whether the guaranteed quality level is provided. Providers need to prove to customers that they fulfill SLA guarantees. This is called SLA validation.

For ***SLA validation*** a wide variety of metrics are important (e.g., delay, loss, jitter). SLA validation requires the measurement of all metrics for which a guarantee is specified in the SLA. The SLA contains a threshold per metric. It is guaranteed by the provider that this threshold is not exceeded. The defined threshold can be for instance a maximum or a mean value (e.g., maximum or mean delay). Furthermore it is important for which time interval the metric is guaranteed. Maximum and mean values are always related to a given interval of time or amount of packets. Nowadays network operators often use a monthly time period. But with increasing demands from applications, there is a strong trend towards much shorter time scales. SLAs on a daily basis are already deployed. SLAs with time intervals of 15 and even 5 minutes are planned.

QoS Metrics can be measured at different layers, ranging from connectivity and signal strength at physical layer up to perceived quality above application layer. In this work only the QoS metrics at network layer are considered.

Passive measurement methods measure exactly the quality that a flow experiences in the network. Due to this they are perfectly suited for SLA validation. The passive measurement of some metrics (like one-way delay) requires the correlation of data from different observation points. That means one has to set up a ***passive multi-point measurement***.

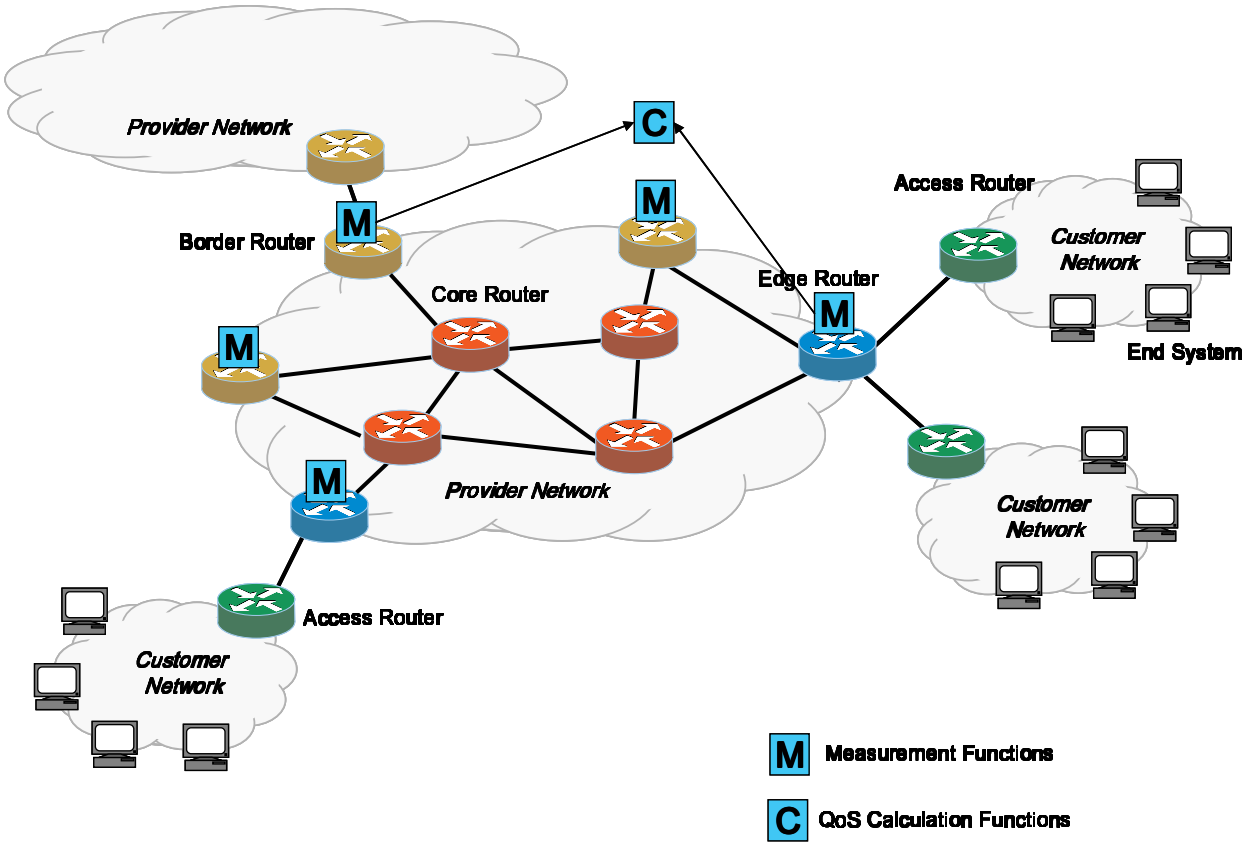


Figure 2-9: Observation Points for SLA Validation

Especially due to the need to transmit per packet information from different observation points (Figure 2-9), the resource consumption for multipoint measurements is much higher than for single-point measurements. Furthermore, the required resources (processing power, transmission, storage) increase with the amount of measured packets. The metric of interest here is the amount of non-conformant packets, i.e., for instance the number of packets that exceed a predefined delay threshold.

3 Sampling for Measurements in IP Networks

Sampling aims at the provisioning of information about a group of elements (the parent population) at a lower cost than a full census of all elements would demand. This is achieved by selecting a finite subset of elements (the sample) and estimating the metric of interest from this subset. The sampling method describes how this subset is selected from the elements of the parent population.

Sampling and estimation methods are a well covered mathematical field. Various methods are described in a wide variety of books (e.g., [Coch72], [Fisz63], [Rinn97], [Schw75]). This work focuses on the applicability of sampling methods to non-intrusive measurement in IP networks. Terms not explicitly defined here are used in the sense of [Rinn97]. The mathematical notation used throughout this document is defined in appendix D.

In this chapter incentives for using sampling for measurements are identified. The main challenges for a usage of sampling in IP networks are described and related work in this area is investigated. Based on this a research plan is developed and the applied methodology is explained.

3.1 The Need for Sampling

The main goal for the usage of sampling methods is the reduction of measurement cost in terms of resource consumption. Measurement functions are mostly supplementary functions, required to ensure the proper operation of the network, monitor resource consumption, validate service specific transmission qualities or detect intrusions. Therefore, measurement costs should be limited to a small fraction of the costs of providing the network service itself. Measurement demands increase due to many reasons:

- ***Increasing data rates:*** Basic measurement functions like capturing, classification, timestamping, and post processing need to operate at higher speeds in order to cope with increasing data rates. Higher data rates also elevate the amount of result data and with this, the resource consumption for processing, storage and result data transmission.
- ***New metrics:*** Future applications may require the investigation of a variety of additional metrics. In past times it was sufficient to observe some key characteristics of network and data transmission like link load or round-trip-times. Nowadays, service level agreements (SLAs), sophisticated accounting methods and increasing security threats (e.g., DoS attacks) require the measurement of much more and different metrics.
- ***Higher granularity:*** The higher granularity (e.g., per flow information instead of link load) that many applications require, additionally burdens functions in the measurement process like classification and storage of per flow information. This problem can become even more complex if IPv6 is deployed. The larger address fields

and the dynamic header structure raise further performance challenges for classification techniques.

- ***Mobility:*** The trend towards mobile communication pushes the deployment of mobile devices and wireless networks. Due to the area of application for mobile devices, their design has to be small, lightweight and energy-saving. Therefore, mobile devices usually have very scarce resources (e.g., processing power, storage). Furthermore, transmission resources in wireless networks are often much more limited than resources in fixed networks. Therefore, the trend towards mobile communication increases the demand for resource efficient measurements.

If one doesn't take care about increasing demand for measurement resources it may happen that resources simply are exhausted. The uncontrolled exhaustion of measurement resources can lead to packet losses in the measurement process (e.g., during packet capturing or transfer of result data) and with this to an unpredictable bias in the measurement results [AmCa89]. Sampling substitutes the uncontrolled discarding of packets by a controlled (random or deterministic) selection process. It reduces the costs for measurement hardware, processing and transmission capacities and prevents the depletion of the available (i.e., the affordable) resources.

3.2 Sampling Challenges

In recent years some solutions have been proposed for volume and packet count estimation. Only few have addressed QoS measurements. The development of new schemes and implementation improvements usually aim at the improvement of the estimation accuracy or a further reduction of required resources.

The first problem in this comparatively new research community is the lack of a common understanding, terminology and categorization of schemes. Terms are often used with different meanings in different publications (e.g., time-based, count-based, event-based). Furthermore, authors invent new names for their schemes (like "smart sampling", "sample-and-hold", "trajectory sampling") or occupy terms with a specific scheme that usually address a broader group of methods (e.g., stratified sampling). This problem was recently addressed by the foundation of a working group on packet sampling (PSAMP) within the IETF. The group works on a common terminology and scheme categorization and will standardize some basic schemes for the use in routers and measurement devices (see [ZsMD05] and [Duff05]).

The appropriate selection of schemes may depend on various aspects of the scenario like the metric that should be estimated, the measurement method, accuracy requirements, available resources, characteristics of the population and the order of processes (i.e., where sampling is applied in the measurement processing sequence). Research results should be assessed only under consideration of the underlying scenario.

An important issue when using sampling schemes is the ***prediction and control of the estimation accuracy*.** Providers need information about the estimation accuracy to provide this information to customers and to asses revenue loss (e.g., if used for accounting). For most applications giving information about the expected accuracy is a prerequisite for customers to accept sampling-based measurements. One can split the challenge into three sub-problems: accuracy calculation, accuracy prediction and accuracy control.

The goal for ***accuracy calculation*** is to find models that show how the accuracy depends on input parameters like sampling settings and characteristics of the population.

Even if a generic model can be found for a sampling method, a further problem is ***accuracy prediction***. For this, it needs to be investigated what information is available before and after the sampling process. It is tried to formulate an accuracy statement only based on the available information.

The ultimate goal for deployment of sampling methods would be ***accuracy control***. If accuracy depends on population characteristics, dynamic characteristics of the population can become a problem, because the accuracy changes permanently. One possibility to address this problem is to adapt sampling parameters to predicted changes of the population from previous samples. Nevertheless, such techniques can contain many sources of errors from estimation and prediction. Furthermore, the control overhead has to be taken into account that is needed to achieve a certain degree of accuracy stability. A permanent re-configuration is certainly not desired, due to the required configuration efforts.

One also could aim at ***prediction and control of resource consumption***. Stable or controllable resource consumption is desired for resource planning and can ensure that resources are neither overloaded nor idle. In the case of unstable resource usage one might need to adapt sampling parameters to available resources. It is likely that this leads to variation of the accuracy and can contradict the targeted fixed accuracy.

A further challenge occurs if more than one measurement point is involved in the measurement. If per-packet information needs to be correlated from different observation points (like in non-intrusive one-way delay measurements), it has to be ensured that the same packets are selected at all observation points. That means one needs to ensure ***synchronization of sampling processes*** at different observation points. Challenges and potential solutions for sampling synchronization are described in section 3.5.

3.3 Taxonomy for Packet Selection Methods

One problem for the assessment and comparison of sampling methods is the lack of a common terminology and a taxonomy for the categorization of data selection schemes for network measurements. In this chapter such a taxonomy is derived under consideration of existing work (presented in chapter 3.6) and current standardization effort. An earlier version of this taxonomy was already presented in [Zseb02]. A formal description of selection

functions that generalizes packet and flow selection can be found in [Duff04]. Parts of this taxonomy have been contributed to standardization [ZsMD05].

3.3.1 A Model for the Description of Packets and Flows

For the description of packet selection schemes packets are considered as the basic elements that form the population. In accordance to [Duff04] the elements of a sequence of packets that arrive at one observation point are described by three characteristics:

- The sequence number[3] *s*, which represents the position of the packet in the sequence of arriving packets.
- The arrival time *t* at the observation point.
- The packet content *c*, which includes packet header and payload.

With this definition, a packet can be represented by the triple $<s, t, c>$. The packet content consists of packet header and payload. An important part of the content is the packet size. In the following the packet size of the i^{th} packet in a flow is denoted with x_i. The whole traffic mix observed within a measurement interval can be described as a sequence of packets.

$$<s_1, t_1, c_1>, <s_2, t_2, c_2>, \ldots <s_N, t_N, c_N> \ldots$$

A set of packets with common properties is called a flow.

In most cases the classification of packets into flows is based on common header fields (e.g., all packets with the same source address). That means the classification rules can be described by a function of the packet content *f(c)*. The attributes (e.g., header fields) that determine to which flow a packet belongs are called ***flow keys*** [Clai05]. The combination of flow keys is called flow identifier or ***flow ID***. A flow can be described by the sequence of packets that belong to the flow.

$$<s_1, t_1, c_1>, <s_4, t_4, c_4>, \ldots <s_8, t_8, c_8> \ldots$$

The number of packets that belong to a specific flow *f* is denoted with N_f. The distance between subsequent packets is called inter-packet distance and can be measured in number of packets or time. From the sequence numbers one can derive how many packets from other flows were observed between the packets of the flow of interest. From the arrival times the time between packets of the flow can be derived. With the knowledge about all three attributes of all packets in the population one could reconstruct the flows observed at an observation point.

Flows usually consist of multiple packets. Therefore flows can have much more attributes than packets. Besides the characteristics of all individual packets a flow can also be

[3] This sequence number denotes the position in the packet sequence of all packets. It has nothing to do with sequence numbers used by protocols (e.g., TCP). In [Duff04] this is called index *i*. In this work the letter *s* is used here instead, because the letter *i* is used as general index for different subsets or populations.

characterized by the distribution of specific packet attributes (e.g., distribution of packet sizes, inter-packet distances) in the flow or aggregated attributes (e.g., number of packets or bytes per flow, mean, variance or other parameters of the distributions). To save resources usually only aggregated metrics are stored per flow (Figure 3-1).

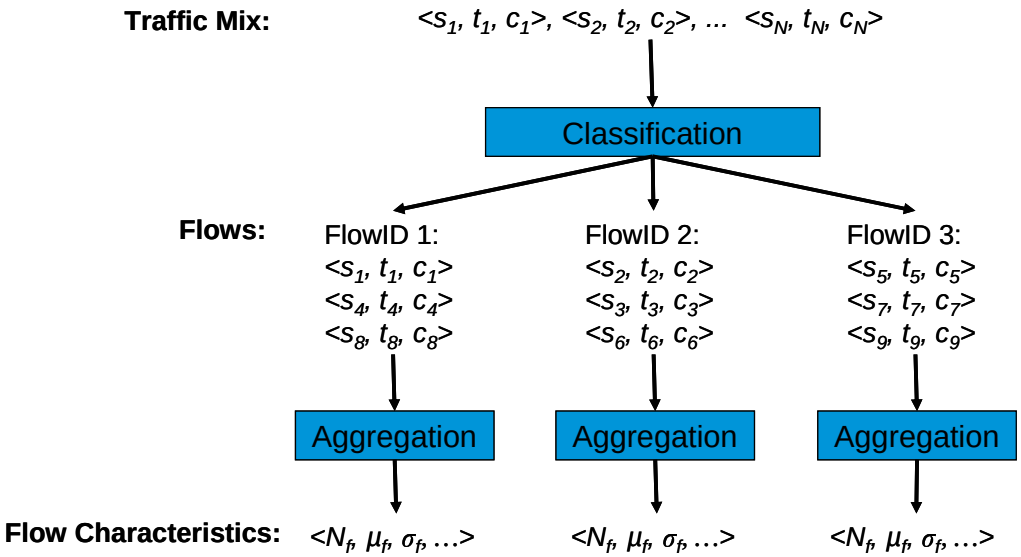


Figure 3-1: Flow Generation

Examples for flow characteristics are:

- The number N_f of packets in flow f
- The flow volume, which is the sum of the packet sizes
- Parameters of the packet size distribution (e.g., packet size mean or standard deviation)
- Parameters of the distribution of arrival times
- The flow duration, which is the difference between the arrival time of the last and the first packet

3.3.2 Packet Selection Methods

Packet selection describes a class of data selection methods that consider packets as basic elements. All IP packets observed in a measurement interval are considered as the ***population*** and the selected packets as the ***sample***. The number of elements in the population is called population size. The number of elements in the sample is called sample size. The target sample size is the number of samples that one wants to select and is denoted with n_T. For some selection methods the real number of selected packets can differ from the target sample size. The real sample size is denoted by n_R.

The ***sample fraction*** is the sample size divided by the population size. As for the sample size it is distinguished between the target sample fraction f_T and the real sample fraction f_R. A deterministic packet selection based on the packet content is called ***filtering***. All other random or deterministic packet selection methods are called ***sampling*** [ZsMD05]. The taxonomy

comprises all packet selection methods, but the remainder of this work concentrates only on sampling methods (Figure 3-2).

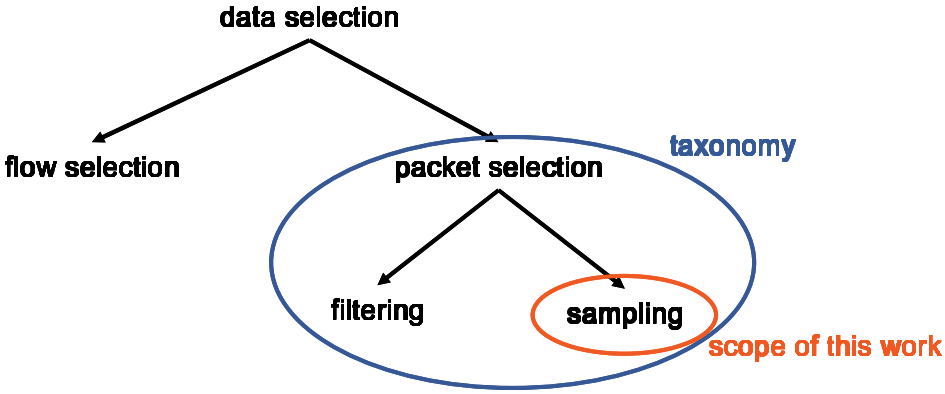


Figure 3-2: Data Selection Methods

Selection schemes can be distinguished in accordance to the following dimensions:

- ***Measurement Interval Definition:*** The measurement interval (MI) defines the interval for which the metric of interest should be calculated and reported.
- ***Selection function:*** The selection function defines how elements are selected from the population.
- ***Input parameters:*** The input parameters are the parameters that are needed by the selection function to make a selection decision.

In addition to the basic characteristics of a scheme it is also possible to concatenate selection schemes and to adapt parameter settings with regard to population characteristics or other external events. Therefore the following additional characteristics are considered for the categorization of schemes:

- ***Number of selection steps:*** The number of subsequent selection functions that are performed to select the elements from the population.
- ***Configuration Dynamics:*** The dynamics of the configuration defines whether configured parameters for the selection function remain constant for all measurement intervals or are modified (e.g., for adaptive methods).

Another data selection method used for network measurements is ***flow selection***. For flow selection flows are considered as basic elements of the population. Although in principle the same methods as for packet sampling can be applied to flow sampling (see e.g., [Duff04]), flows have different attributes than packets. Due to this, the potential input parameters for the selection functions differ. Since flows consist of multiple packets there are much more characteristics that could be considered for a selection of flows than for a selection of packets (e.g., distribution of packet sizes or arrival times, proportion of packets of a common type, etc.). Furthermore, it depends on the implementation which flow attributes are stored. The IPFIX information model [QuBM05] defines a variety of flow attributes that should be

reported in flow records, but allows vendor specific variations. That means the same dimensions can be used to classify flow selection methods, but the variety and type of parameters would differ. Due to these reasons, flow selection is not considered in this work and not part of the taxonomy.

3.3.3 Measurement Interval Definition

The measurement interval (MI) defines the interval for which the metric of interest should be calculated and reported. Therefore the MI defines the population for the selection process. There are different ways to define a measurement interval:

- ***Count-based definition:*** The measurement interval is defined in number of packets. With this definition the number of packets (size of the population) is fixed but the time duration of the MI is variable.
- ***Time-based definition:*** The measurement interval is defined as time interval. With this the duration of the interval is fixed but the number of packets within the MI is variable. That means that the size of the population varies.

Both approaches have advantages and disadvantages. A count-based definition ensures a constant population size for subsequent MIs. On the other hand their time duration may be very long if packet rates are low. The reporting of measurement results at certain time-periods cannot be ensured.

Time-based MIs ensure a reporting at given time intervals, but they may contain only few or even no packet if only few or no packets are observed within the time interval. That means the population size varies for subsequent intervals.

Mixed interval definitions are also possible. One can for instance start the measurement interval at a specific time and end it after a fixed amount of packets was received. Measurement interval definitions could be also based on the occurrence of specific events (e.g., arrival of a specific packet type). Such definitions are not considered in this work.

3.3.4 Selection Function

The selection function defines the rules for the selection of elements. It makes a selection decision for each element of the population based on input parameters and selection rules. In accordance to [ZsMD05] the following two basic selection algorithms are distinguished:

- ***Random selection:*** The packet selection is based on a random function.
- ***Systematic selection:*** The packet selection is based on a deterministic function.

A random selection can be performed by two different basic algorithms. In ***n-out-of-N*** sampling exactly *n* elements are selected from the population, which contains *N* elements. For ***probabilistic*** sampling each element is selected with a specific probability *prob*. The selection decision is made independently for each packet. Therefore the number of selected elements can vary in probabilistic sampling.

Random probabilistic schemes with a fixed selection probability per element are called ***uniform probabilistic*** schemes. If the selection probability depends on packet attributes (e.g., packet count, arrival time or content) they are called ***non-uniform probabilistic*** schemes.

3.3.5 Input Parameters

Input parameters represent the information that is needed by the selection function to make a selection decision. One has to distinguish between different input parameters for the selection process.

- ***Configuration parameters*** are the parameters of the selection process that are pre-configured (e.g., number of elements that should be selected).
- ***Characteristics of population elements*** are the parameters which are directly related to elements of the population (e.g., packet arrival time).
- ***External events*** are input parameters that are not pre-configured and are not directly related to the population. (e.g., device state).

The required configuration parameters for a scheme are scheme specific and defined by the selection function. Configuration parameters for basic schemes can be found in [ZsMD05]. Examples for schemes that use external events as input parameters are router-state and flow-state dependent schemes introduced in [ZsMD05]. Such schemes are not considered here.

For the categorization of packet selection schemes it has to be considered which attributes of the packets are relevant for the packet selection. The selection decision can depend on the following packet attributes:

- For ***count-based*** schemes the selection decision is based on the packet count, i.e., the packet position in the sequence of packets (e.g., selection of every 10th packet).
- For ***time-based*** schemes the selection decision is based on the packet arrival time (e.g., selection of all packets that arrive in a predefined time interval).
- For ***content-based*** schemes the selection decision is based on the packet content (e.g., selection based on the result of a hash function over the packet content).

Based on the packet attribute of the arriving packet it is decided whether a specific packets is selected or not. Since the packet attributes arrival time, packet count and packet content cause the selection of a packet from the population, they are in the following also called ***trigger*** for the selection process. The trigger can be seen as an input parameter for the calculation of the selection probability per packet. A special case is uniform probabilistic sampling. Since the selection probability is the same for each packet in the measurement interval, it is independent of any packet attribute.

It is possible to combine schemes. One example for a combined scheme would be to start sampling every minute and then select the next n observed packets. This scheme has a time-based start trigger and a count-based stop trigger.

3.3.6 Definition of Basic Packet Selection Schemes

Basic schemes can be defined by the dimensions that were identified above.

- Measurement Interval (MI) definition (tine, or count-based)
- Trigger for the selection in Scheme (time, count, content)
- Selection method (random, systematic)

All basic schemes comprise of a single selection function and have a static configuration. The schemes are indicated as follows:

MI-Definition/Trigger/Selection

Where the following notation is used: T-time-based, C-count-based, Co-content-based, RP-random probabilistic and RN-random n-out-of-N. That means for example T/C/RP stands for time-based measurement interval definition, count-based trigger, and random probabilistic selection. A random n-out-of-N selection with a time- or content-based trigger is not applicable, because it cannot be ensured that exactly *n* packets are selected if the selection depends on arrival time or content. Table 3-1 provides an overview of the basic schemes with examples.

MI	**Trigger**	**Selection**	**Description**
T	-	RP	Uniform probabilistic with time-based MI *Example*: Select each packet with a fixed probability.
T	T	S	Time-based systematic sampling with time-based MI *Example*: Start every 60 seconds a sample interval of 10 seconds. Select all packets that arrive within the sample intervals (i.e., packets for which the arrival time is part of the interval)
T	T	RP	Time-based random probabilistic sampling with time-based MI *Example*: Select all packets that arrive in a given time interval with probability A and all others with probability B.
T	T	RN	NOT APPLICABLE
T	C	S	Count-based systematic sampling with time-based MI *Example*: Select every 10^{th} packet in the MI.
T	C	RP	Count-based random probabilistic sampling with time-based MI *Example*: Select each packet in the MI with a given probability.
T	C	RN	Count-based random n-out-of-N sampling with time-based MI *Example*: Select exactly *n* packet from all *N* packets in the MI.
T	Co	S	Filtering *Example*: Select all packets in the MI that have a specific source address.
T	Co	RP	Content-based random probabilistic sampling with time-based MI *Example*: Select all packets with a specific source address with probability A and all packets with other source addresses with probability B.
T	Co	RN	NOT APPLICABLE
C	-	RP	Uniform probabilistic with count-based MI *Example*: See T/-/RP

MI	Trigger	Selection	Description
C	T	S	Time-based systematic sampling with count-based MI *Example*: see T/T/S, but count-based MI definition leads to difficulties
C	T	RP	Time-based random probabilistic sampling with count-based MI *Example*: see T/T/RP, but count-based MI definition leads to difficulties
C	T	RN	NOT APPLICABLE
C	C	S	Count-based systematic sampling with count-based MI *Example*: Select every 10th packet
C	C	RP	Count-based random probabilistic sampling with count-based MI *Example*: Select each packet in the MI with a given probability.
C	C	RN	Count-based random n-out-of-N sampling with count-based MI *Example*: Generate n uniform distributed random numbers between 1 and N. Select all packets that arrive at the n selected packet positions.
C	Co	S	Filtering *Example*: see T/Co/S
C	Co	RP	Content-based random probabilistic sampling with count-based MI *Example*: see T/Co/RP
C	Co	RN	NOT APPLICABLE

Table 3-1: Overview of Basic Packet Selection Schemes

For schemes with time-based trigger and time-based MI there are two different possibilities to calculate the statistical inference: based on time-intervals or based on number of packets. If the inference is done based on time intervals, the number of selected time intervals is extrapolated with the number of all time intervals of the given size that would fit in the measurement interval. For an inference based on the number of packets the number of selected packets is extrapolated with the number of all packets in the MI.

There are some difficulties if the dimension of the measurement interval definition and the trigger differ. In time-based MIs the number of packets in the MI varies. That means that for a count-based systematic and random probabilistic sampling also the number of selected packets varies. A count-based n-out-of-N sampling is only possible, if the number of packets in the interval is known in advance. If the MI definition is count-based, the time duration can vary. Using a time-based trigger requires information about the MI duration in order to ensure that selected time intervals do not lie outside the measurement interval.

Figure 3-3 shows an initial assessment of the basic schemes. The x-axis shows the information that is required about the arriving packets in order to perform the selection. The y-axis describes the effort that is needed to perform the selection like the generation of random numbers or the calculation of a selection probability per packet.

The resources required for making a selection decision depend on the effort and on the information that has to be extracted for each packet. Therefore costs increase for schemes that have higher effort or require more traffic information.

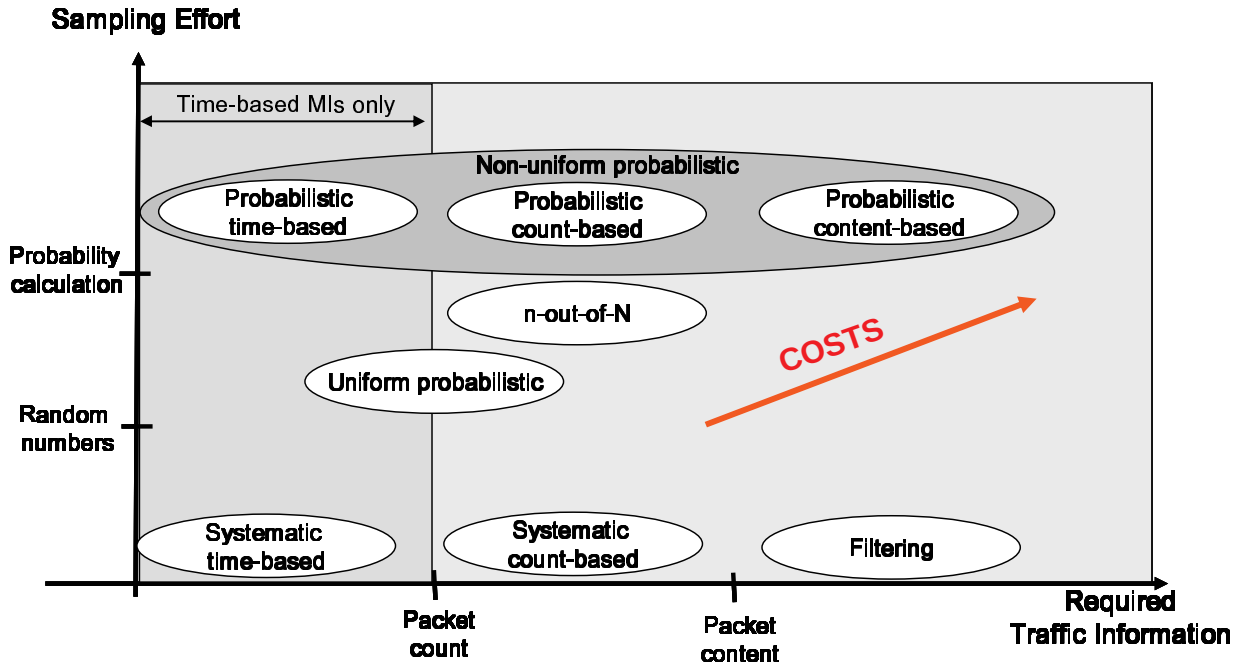


Figure 3-3: Basic Selection Schemes

Systematic time-based sampling can be realized by periodically enabling/disabling the packet capturing function. It requires no traffic information and the effort for realizing the scheme is small. For systematic count-based sampling a packet counter is required. For filtering (systematic content-based selection) one needs access to the packet content.

For uniform probabilistic sampling each packet is selected with a fixed probability. The sampling effort is higher, because it requires the generation of random numbers. A packet counter is needed for the scheme if the extrapolation is done based on the number of packets in the MI. If the extrapolation is done based on time-intervals the scheme does not need a packet counter. Since a sampling decision is required per packet it is likely that the number of packets in the measurement interval is counted anyway.

n-out-of-N sampling requires a packet counter. It is quite similar to uniform probabilistic sampling. It requires slightly more effort than uniform probabilistic sampling because usually a list of n random numbers needs to be maintained.

For uniform probabilistic sampling the selection probability is fixed. For non-uniform probabilistic schemes the probability depends on packet attributes. Therefore those schemes require additionally a calculation of the selection probability per packet based on its attributes.

3.3.7 Multi-Stage Methods

The number of selection steps is a further criterion to distinguish packet selection methods. Single-stage methods select all elements of the sample in one single step. In multi-stage methods the selection of elements is done in multiple steps. It is distinguished between real multi-stage methods and pseudo multi-stage methods. In real multi-stage methods in each stage a selection of elements is performed. Pseudo multistage methods contain at least one

steps were the whole population is processed. More information about multi-stage methods can be found in [Rinn97] and [Coch72]. In this work I focus on stratified sampling method. Stratified strategies and relevant parameters for stratified schemes are described in chapter 3.4.

3.3.8 Adaptive Sampling

In adaptive selection methods the configuration for the selection function is adapted to characteristics of the population or external events (e.g., device state). Due to the dynamic characteristics of most traffic flows it is nearly impossible to find static sampling parameters that are optimal for the whole measurement duration. Adaptive sampling methods use the trace characteristics from past measurement intervals to adjust the sampling parameters of the current measurement interval (Figure 3-4). The adaptation rules define how the sampling parameters are derived from the trace characteristics in previous measurement intervals.

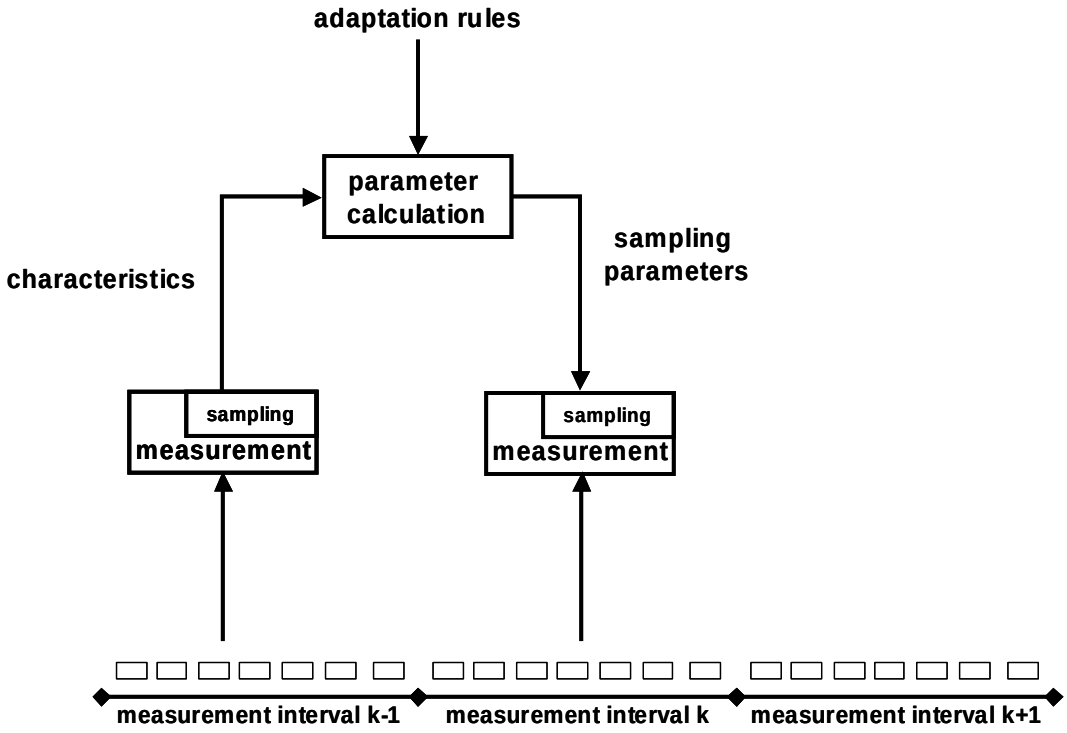


Figure 3-4: Adaptive Sampling

Adaptive sampling can be used to control the accuracy or the resource consumption of a measurement process. It can keep the outcome of the sampling process more or less constant despite the changes in the traffic flow. The estimation accuracy that can be achieved with a given sample fraction usually depends on the variance of the survey variable in the population. To maintain a constant accuracy the sample fraction needs to be adapted to this expected variability. A higher sample fraction is required if it is expected that the metric of interest varies a lot within the measurement interval. It can also be used to adjust parameters towards optimization of resource consumption (e.g., fixed number of sampled packets, fixed number of flow entries).

3.4 Stratified Sampling

Stratified sampling is a pseudo multi-stage method that consists of a classification of the whole population and a selection process (Figure 3-5). In classical stratified sampling the classification is done before the sampling. If the classification is done after sampling, one speaks about post stratification.

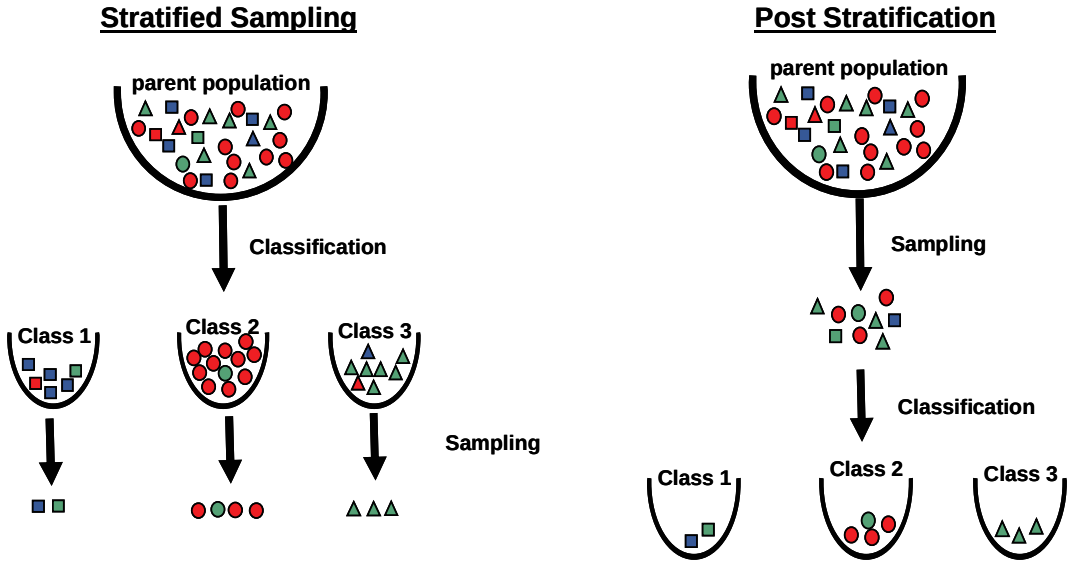


Figure 3-5: Stratified Sampling

The basic idea behind stratified sampling is to increase the estimation accuracy by using a-priori information. The a-priori information is used to perform an intelligent grouping of the elements of the population. The elements of the population are grouped into subsets (strata) in accordance to a ***stratification variable*** *x*. This grouping can be done in multiple steps. Then samples are taken from each subset in order to estimate the ***survey variable*** *y*, i.e., the metric of interest.

The key for increasing the estimation accuracy with stratification is to select a stratification variable that has some correlation with the survey variable. The stronger the correlation between the stratification variable and the survey variable, the easier is the consecutive selection process. If the stratification variable were equal to the survey variable, each element of a stratum would be a perfect representative of that characteristic. In this case it would be sufficient to take one arbitrary element out of each stratum to get the actual distribution of survey variable in the population. Therefore stratified sampling can reduce the number of samples needed to achieve a given accuracy significantly. The difference between the accuracy that can be achieved with a stratified scheme compared to a single-stage random sampling is called the ***stratification gain***.

A stratification strategy is defined by the following parameters:

- ***Stratification variable:*** What characteristic(s) of the population elements are used to group the population?
- ***Number of strata:*** How many strata should be defined?

- ***Strata boundaries:*** Which strata boundaries should be used?
- ***Allocation method:*** How should the number n of samples be distributed over the strata

The steps are explained below. A comparison of different stratification strategies and rules to set stratification boundaries can also be found in [Zseb03].

Also post stratification methods are of interest for the application of sampling techniques to measurement, because it reduces the number of elements that needs to be classified and with this unburdens the classification process. Unfortunately, in the measurement case the number of elements that belong to one class is usually unknown. Therefore one lacks an important piece of information for the inference of population characteristics from the sample. The estimation of per class characteristics from classified samples without knowledge about the class proportions in the parent population is a quite complex problem that is addressed in chapter 4.

3.4.1 Stratification Variable

A stratification gain can only be achieved if the stratification variable has some correlation to the survey variable. Furthermore, the characteristic that is used for the stratification is needed for each element of the population. Therefore it should be known for each element in advance or it should be easy to obtain it from each element of the population (at least easier that the survey variable). Otherwise the effort to perform the stratification grows too high. Therefore a suitable stratification variable has the following two properties:

- The values of the stratification variable are correlated with those of the survey variable.
- The values of the stratification variable are easier to obtain than those of the survey variable.

3.4.2 Number of Strata

The question how many strata are suitable tightly corresponds to the question whether a benefit from stratification with the chosen variable can be expected at all. If it turns out that defining just one stratum leads to similar results than defining more strata, the chosen characteristic is not appropriate to be used as stratification characteristic.

The achievable gain depends on the correlation between the survey and the stratification variable and on the number of strata. Defining more strata usually only increases the gain a little bit. Furthermore, the stratification effort increases if more strata are used. [Coch72] showed that in most cases it is not advisable to use more than six strata.

3.4.3 Stratification Boundaries

If the number of strata is determined, it still needs to be defined where the boundaries of the strata should be set. The goal is to define strata in a way that the variances of the survey variable within the strata are as small as possible. The best basis for setting the strata

boundaries would be the distribution of the survey variable itself, but this is not known. Based on the assumption that the relation between the stratification variable and the survey variable is more or less linear, one uses the distribution of the stratification variable as basis for the boundary calculation instead.

The simplest boundary definition method is to just divide the x-scale into (more or less) equal intervals. Here one can distinguish between using the theoretical range of x values and using the actual measured range of x value.

A further approximation procedure for defining stratification boundaries is the cumulated $\sqrt{f}$ method proposed in [DaHo59]. This rule of thumb says that strata boundaries should be set in a way that one gets equal intervals on the cumulated $\sqrt{f}$ scale, where *f(x)* denotes the frequency distribution of the stratification variable *x*.

3.4.4 Allocation Methods

The allocation method defines how the number of samples should be distributed over the strata i.e., how many packets are to be selected per stratum. The simplest allocation method is the ***equal allocation***, where the same number of elements is selected from each stratum, regardless of the number of elements in the stratum.

$$n_l = \frac{n}{L} \tag{3.1}$$

This method is not very efficient if there are large differences in the amount of elements in the strata. A better method is the ***proportional allocation*** where the number of samples n_l per stratum l is chosen proportional to the number of elements N_l in the stratum.

$$n_l = n \cdot \frac{N_l}{N} \tag{3.2}$$

This method is suitable if the variances within the strata are of a similar magnitude.

A further improvement can be achieved if the (expected) variance of the survey variable for each stratum is taken into account for the allocation. It is easy to understand that it is better to select more elements from a stratum with a high variance of the survey variable than from a more homogeneous stratum. Furthermore, it has to be taken into account that the costs for obtaining the needed information from elements in one stratum may differ from the costs that occur per element for another stratum. An allocation scheme that takes this into account is the ***optimal allocation***. A special case is the Neymann-optimal allocation where it is assumed that the investigation costs per element are the same for all strata. The amount of samples per stratum for the Neymann-optimal allocation is calculated as follows:

$$n_l = n \cdot \frac{N_l \cdot \sigma_l}{\sum_{l=1}^{L} N_l \cdot \sigma_l} \tag{3.3}$$

This method provides the best allocation strategy (if costs per element are stratum independent) and is especially suitable in cases where the stratum variances differ much.

Nevertheless, the optimal allocation also has a few disadvantages. First of all, it requires knowledge about the variances within the strata. Since these variances are unknown they need to be estimated from previous surveys. Furthermore, it can happen that the calculation of n_l in accordance to the optimal allocation results in a higher number of samples than number of elements in the stratum. Due to these reasons the proportional allocation is more often used than the optimal allocation and also chosen in this work. More information about setting stratification boundaries and allocation schemes can be found in [Zseb03].

3.5 Sampling Synchronization

As seen in 2.4.4 some measurements require the correlation of data from different observation points. It has to be ensured, that all packets of interest are captured at all involved observation points. In order to calculate a delay value for one packet, the same packet has to be captured and uniquely identified at both observation points. This is already a problem if no packet selection is deployed, because packets do not necessarily take the same path and can get lost, duplicated, reordered on the way from source to destination. Furthermore, flows from other sources that arrive at the same destination interfere with the flows of interest and lead to a different traffic mix at different observation points.

If packet selection is used, one has to ensure that the same packets are selected at the observation points. Due to the different arrival sequence and arrival times at different observation points, this is usually not possible even if the same sampling scheme is applied with the same configuration parameters and the same random seed.

One possibility to cope with this problem is to use a heterogeneous measurement infrastructure. In this one or more reference points are provided at which all packets are measured. The data collection and metric calculation should be co-located or close to this reference point in order to prevent the transmission of all packet information. Packet selection can be applied at other observation points that report to this reference point. With this the effort to measure all packets is only required at the reference point. The data that needs to be transferred from the other observation points is reduced by the packet selection methods. Reference points could be a few high-speed meters that are located close to servers or other main communication nodes.

Another option is to use filtering, which is a content-based systematic packet selection (see 3.3). Most bytes of the packet content (packet header and payload) remain the same during transport. If only those fields are considered for the filtering, a selection of the same packets at different observation points is ensured. The problem with filtering is that it is a systematic scheme and therefore biased towards the packet content that is used as selection criteria in the filtering rules.

One special form of filtering is a hash-based selection. For this a hash-function is applied to the selected bytes of the packet content. If the hash value falls in a predefined range, the packet is selected. In [DuGr00] it is shown how a hash-based selection can be used to emulate

a pseudo-random probabilistic sampling. If a sufficient amount of the IP packet header and payload were used as input to the function, the bias with regard to specific packet attributes was only small for the investigated traces. Further tests on the uniformity of the hash range (output values of the hash function) for different hash functions were performed in [MoND05]. Nevertheless, the evaluation results are very specific to the investigated functions, attributes and traces. They cannot simply be generalized to arbitrary traffic traces, other packet attributes or different hash-functions. Furthermore a hash-based selection requires the processing of each packet and therefore is very resource-intensive compared to other methods.

3.6 Sampling: State of Art

Growing packet rates on high speed links and limited resources for capturing, storage and post-processing triggered the first interest in sampling methods for packet count and volume estimations. Already in 1989 Paul Amer and Lillian Cassel proposed to use sampling for real-time status reporting in IP networks **[AmCa89]**. The paper describes sampling-based measurements of ***peak load, packet rates*** and changes in those metrics. Two different sampling techniques are introduced, a systematic and a random method with a time-based start trigger and a count-based stop trigger, realized by the enabling and disabling of the receive function of the measurement device. The paper provides an introduction to the problem space and shows exemplarily how statistical concepts like parameter estimation and statistical testing can be used to detect changes in network load. The paper does not contain any practical tests or validation of results.

An often cited early work on sampling was published in 1993 by K.C. Claffy **[ClPB93]**. She and her co-authors describe the empirical investigation of different sampling schemes for the estimation of ***distributions of packet sizes*** and ***interarrival times***. They consider the whole packet stream at the observation point, so no classification is done here. Goodness-of-fit tests are used to compare the distributions derived from the sampled packets with the original distribution of the population. In order to compare results from different sample sizes a variation of the chi-square test, the phi coefficient, was used to measure the degree of similarity between distributions. Five different sampling schemes are compared: Count- and time-based systematic sampling, count- and time-based stratified sampling and a count-based n-out-of-N sampling. Stratified sampling here uses the time and packet count as stratification characteristics. In order to compare the sampling methods, offline tests are performed with a one-hour trace of packets collected at the FDDI (Fiber Distributed Data Interface) entrance interface from the San Diego Supercomputer Center (SDSC) into the NFSNET (National Science Foundation Network) backbone San Diego. The sampling schemes are applied to the trace and results are compared with the real distributions of the whole population. Although the paper describes the applicability of goodness of fit tests, no theoretical consideration or modeling of potential estimation errors is made. The assessment of sampling schemes is based

on empirical investigations only. In her experiments count-based schemes performed better than time-based schemes. But since only one trace was investigated, the empirical results are not sufficient to justify general statements about the performance of the investigated schemes.

Since many applications require the measurement of per flow characteristics, methods evolved that allow an estimation of packet counts and volumes per flow.

3.6.1 Flow Sampling and Attention to Heavy Hitters

For flow measurements the measurement process contains classification and sometimes also aggregation functions. Therefore it is relevant for resource consumption and estimation accuracy, where in the processing chain packet selection and flow selection functions are applied.

With more fine grained flow definitions, the number of flows increases and the enormous number of flows becomes a problem. Even if traffic is sampled on packet level the number of flows can remain high. If only one packet from a flow is sampled, an entry needs to be created and maintained in the flow cache, leading to considerable processing and memory demands.

Therefore some recent research concentrates on estimating the packet count and volume for ***heavy hitters*** (few large flows that carry most of the traffic). Those approaches are aimed at a small or stable memory and transmission resource consumption. This can be achieved either by neglecting or discarding flow entries for small flows (***flow sampling***) or by biasing the packet selection towards large flows. In flow sampling the selection is done after packets were classified into flows or even after flow records have been generated that do not include the individual packet attributes (e.g., size of each packet) but only flow attributes (e.g., number of packets, total volume of flow, etc.)

A simple method to combine packet and flow selection was already introduced in **[JePP92]**. The paper describes the estimation of ***packet counts*** that belong to a specific source. A count-based probabilistic sampling is applied to the data stream. Only for the selected packets it is examined from which source they originate. That means that the classification is applied after the sampling process. A flow list is maintained where for each source the corresponding packet count is stored. If a new packet is sampled and the list already contains the source, the counter for this specific source is increased. If the packet belongs to a new source (no entry exists in the list) a new entry is created. The memory demands are kept constant by limiting the flow list. If the list becomes too large, the smallest entry is removed. With this method, early removals and partial counts can lead to an incorrect ranking of flow entries. The authors show that with some assumptions about the traffic characteristics, such miscounting errors are negligible and how the packet counts of the t largest sources can be estimated if t is small. For investigations on the estimation accuracy both, the data stream and the sampling process are modeled as discrete-time stochastic processes.

In **[DuLT01]** a ***flow sampling*** scheme is introduced tailored for usage-based accounting. The proposed method allows a biased selection of flows dependent on their sizes for the estimation of the total traffic on a link. Large flows are selected with a higher probability. This biased selection is motivated by the heavy-tailed distribution of packets and bytes sizes per flow, i.e., few large flows contribute most of the traffic. This phenomenon is observed in many traffic traces (see section 2.1.2). The presented sampling method is called ***smart sampling***[4]. It is a non-uniform probabilistic flow selection. The method is based on a sampling probability function which defines the selection probability as a function of the flow size. The selection probability provides the input parameter for a probabilistic sampling that is applied to the generated flow records. The method is further elaborated and compared to other techniques in later publications ([DuLT04], [DuLT05a]). [DuLu03] investigates the combination of this flow selection method together with a probabilistic packet selection in the router and the unintentional dropping of flow records on the way to the collector due to transmission errors.

Christian Estan and George Varghese propose a method called ***sample-and-hold*** **[EsVa01, EsVa02a-b, EsVa03]**. First all packets that arrive at the router are classified with regard to the flow keys in order to find out to which flow they belong. So the classification is done before the sampling. Then it is checked whether there already exists an entry for this flow. If an entry exists the packet and byte count for this flow is updated with the received packet. If the packet belongs to a new flow and therefore no entry exists for the flow, the packet is forwarded to a probabilistic sampling process. There it is decided with probability p whether the packet is selected for further processing or not. If it is selected by the sampling process, a new entry is created for this flow. If it is not selected, the packet is discarded and no new flow entry is created. This method realizes a biased flow selection that concentrates on the large flows. Although a packet sampling scheme is applied to achieve the flow selection, this method can be considered as flow sampling scheme. The sampling process is applied after classification before flow record creation. This unburdens the flow cache memory and the flow record transmission but packet classification has to be done at link rate. It reduces the memory consumption (in the flow cache) and the required transfer resources for flow records but does not (as other approaches like NetFlow) unburden the classification process.

The authors of **[MoUK04]** propose a technique to ***identify elephant flows*** (flows with large number of packets) from sampled packets. They use a flow definition based on the 5-tuple consisting of source and destination address, source and destination port, and protocol. Their definition considers a flow as elephant flow if it contributes to more than 0.1% of the traffic. The paper mainly describes how a threshold can be found for identifying a flow as elephant based on the number of packets of that flow in the sample. The authors also show how an

[4] In a later paper [DuLT05] this methods is termed threshold sampling.

appropriate a priori distribution can be chosen. Although the theoretical considerations are based on probabilistic random sampling, the authors claim that the scheme also works with systematic sampling, because multiple flows interleave. For the investigated trace of 10^7 packets (137 seconds), periodic sampling performed similar to the theoretical expected results for random sampling. Nevertheless, the assumption that periodic sampling on a traffic mix leads to random sampling per flow is based on an unproven statement in [DuLT03], and has not been verified in general in this paper.

In **[KoLM04]** the ***packet rates*** (packets/second) of the flows on a link are estimated, based on the measured link rate and random samples from the traffic mix. The authors argue that link rates are easy to measure whereas sampling methods are needed to reduce the measurement overhead (i.e., classification efforts per packet) for per flow measurements. A specific packet selection technique, called ***RATE (Runs bAsed Traffic Estimator)***, is used to bias the selection towards large flows (flows with many packets) and with this reduce memory consumption for the flow cache. The methods works by detecting so-called two-runs, instead of keeping an entry for every flow. A two-run is the occurrence of two subsequent packets from the same flow which arrive directly after each other. If a packet arrives it is first classified to which flow it belongs. Then the flow ID is stored. If the next arriving packet has the same flow ID a two-run for the flow has occurred and the two-run counter for this flow is increased (or a new entry is created if this flow has not had a two-run before). If the next arriving packet has a different flow ID than the previous packet, the flow ID of the new packet is stored and the previous packet is discarded. That means the packet selection method is applied after classification and therefore does not unburden the classification process. According to PSAMP terminology the applied selection method is a ***filtering*** technique because it is a deterministic selection based on packet content. Furthermore, since the filter criteria change with packet arrival dependent on the last packet arrival from the flow the method is a ***flow-state-based filtering*** with regard to PSAMP terminology.

A different approach is followed in **[KuXW04]**. The authors aim at estimating the ***packet count*** for all flows, regardless of their size. They propose a novel data structure, the ***Space Code Bloom Filter (SCBF)***, which allows operation at OC-192 and higher. The updating of packet counters per flow is done based on groups of independent hash functions that increase a counter if packets from the flow have been observed before. The update does not require any read operation and therefore can operate at very high speeds. For the single SCBF approach there is no explicit packet selection. Each arriving packet is processed. Nevertheless, with limited memory, packets from different flows may hash to the same location. Therefore the method does not return the exact packet count. The loss of information here is caused by a lossy data structure. Nevertheless, the authors show that the packet count can be estimated with a good accuracy by using a maximum likelihood estimation on the stored information.

3.6.2 Adaptation to Accuracy Requirements and Resource Limitations

First adaptive approaches aimed at a ***constant resource consumption***. In **[DrCh98]** an adaptive sampling scheme is presented that aims at a better utilization of available CPU power. The basic idea is to adapt sampling parameters to the available processing power. For this the authors extend a static systematic count-based scheme presented in [ClPB93]. Furthermore, in contrast to [ClPB93], which considers the whole data stream, they apply a flow classification based on source addresses. In their example the authors reserve 50% of the CPU power for the packet processing. If the incoming packet rate is low many or even all arriving packets can be processed without exhausting the processing power. If the packet rate is high, the sampling algorithm selects only the amount of packets that the CPU can process. With their method they estimate ***packet count mean***, ***packet count variance***, ***peak-to-mean ratio (PMR)***, and the ***Hurst parameter***. The Hurst parameter provides a measure of the self-similarity of traffic and serves as an indicator of traffic burstiness. The better utilization of the available CPU power allows to sample more packets in times of low incoming packet rates. The authors show by empirical experiments that with adaptation their method provides more accurate estimates.

Further approaches aim at a ***stable estimation accuracy***. Baek-Young Choi introduces an interesting concept of adaptive sampling for the detection of changes in traffic load **[ChPZ02a-c]**. She uses time-based measurement intervals and a random probabilistic[5] sampling for the packet selection. The goal here is to keep the estimation accuracy (error and confidence level) in given boundaries. This is achieved by adapting the sampling probability within an observation period to the expected traffic characteristics of that interval. In order to keep the estimation accuracy within the given boundaries, two steps are performed. First the squared coefficient of variation (SCV) of packet sizes in the current observation interval is estimated from the samples taken from that interval. The SCV is the variance of the packet sizes in the observation period divided by the square of the mean value for that interval. This estimate is then used to predict the SCV value of the subsequent interval. Since time-based measurement intervals are used, the number of packets per measurement interval varies. Therefore also the number of packets observed in that period is counted and used to predict the packet count of the next observation period.

The performance of the scheme is empirically investigated. For this, the adaptive scheme is applied to real traffic traces (e.g., from the Auckland-II trace collection). The results of the adaptive scheme are compared to optimal sampling, where the sampling probability is calculated error-free with the real SCV and packet count of the block (which in reality is

[5] The paper does not explicitly state that probabilistic sampling is used, but since the selection probability is adapted one can assume the use of probabilistic sampling.

unknown) and to static random sampling without parameter adjustment. It is shown that the adaptive scheme provides stable accuracy boundaries in contrast to the static scheme and performs nearly as well as the error-free optimal sampling. Furthermore it is shown that for the investigated traces the SCV decreases if the load increases. That means especially in times where link utilization is high one can estimate the load based on less samples while maintaining the same accuracy boundaries. The adaptive sampling approach looks quite promising. Nevertheless, it has to be investigated at which time scales a change of sampling parameters is possible and reasonable. Furthermore the costs of re-configuration should be considered (e.g., overhead for re-configuring of the sampling process). In addition to this for probabilistic sampling the sample size varies, leading to a variable resource consumption. Dynamic changes of the sampling probability amplify this effect.

The authors of **[EsKM04]** propose to extend NetFlow to allow an adaptation of the sampling rate. This is motivated by the fact that the resource consumption, memory and bandwidth required to store and transport flow records, highly depends on the number and type of arriving packets at the observation point. It is especially pointed out that the flow cache requirements and the number of flow records can grow immense when a flooding attack hits the router, introducing additional load problems for the router under attack. This can be prevented by using adaptive NetFlow which reduces the sampling rate if the packet rate increases. Although NetFlow uses a 1-in-K sampling the modeling in the paper is done for probabilistic sampling.

They introduce the concept of using fixed time bins for the flow reporting and propose that users should define what number of flow records they would like to get for each measurement interval. The sampling rate is then adapted in accordance to this desired number of records in order to keep the resource consumption constant. If the packet rate is low, a higher sampling rate can be used to get a higher accuracy. If the packet rate is high, for instance because an attack is in progress, the sampling rate is reduced to not exhaust the available resources.

3.6.3 Sampling Side Effects and Further Metrics

Few has been published on ***side effects*** of the deployment of sampling. In **[DuLT02]** Nick Duffield addresses three different issues regarding sampling. Influences on flow reporting, on resource consumption and on the accuracy for volume estimations are investigated. In the first section it is investigated how sampling influences the reported number and duration of flows. Main effects occur due to interrelations of the sampling process with flow termination criteria. A flow is for instance considered as finished if no packets are observed for a predefined time interval. If only some selected packets of the flow are reported, the flow might be considered as over although packets arrive in the time interval, just because those packets were not selected. In order to estimate how this effect influences the resource consumption a mathematical model is defined to estimate the number of reported flows and the mean number of flows that are active in the memory. For this a very simplified scenario is considered with

only one single flow, with equally spaced packets and systematic (count-based, equally spaced) sampling. This simple model then is generalized to a case with multiple flows by summing up the average numbers for one flow. With this one can calculate an estimate for the total number of reported flows and the mean number of flows that are active at the same time in the flow cache. With empirical investigations (but only with one single trace) it is shown that the predicted values for the total number of flows and the mean number of active flows lie within the values that occur if real sampling is done on the investigated trace with an error of ±10%. Since no mathematical model for the accuracy calculation is given and only one trace was investigated, these results cannot be generalized. That means it still remains unclear what accuracy would be achieved with other traces. In the third section of the paper it is investigated how flow characteristics can be estimated from the sampled packets for probabilistic sampling. Three metrics are of interest: bytes and packets for the traffic mix and per flow, total number of flows and the average length of flows. The number of packets and bytes of a specific flow can be estimated with the method of moments. It is shown how the estimation can be assessed by the variance of the estimate.

In recent years researchers used sampling also for the estimation of more unusual metrics like temporal characteristics or spectral density of the packet arrival process or the tracking of the path a packet takes through the Internet.

The authors of **[PASF02]** focus on the detection of ***temporal correlations*** in a trace. They introduce a sampling method called Fast Correlation-Aware Sampling (FastCARS). The method consists of a superposition of multiple systematic count-based sampling processes (in the paper this is called deterministic event-driven sampling). The authors show how other systematic count-based sampling schemes can be formulated as special cases of the FastCARS method. They perform empirical tests on different traces from the NLANR data set [NLTraces]. The experiments show that the proposed method provides better results for estimating interarrival time distributions than simple count-based methods used in [ClPB93]. When exploring the independence of interarrival times they discovered that interarrival times are not independent. The occurrence of packet trains [JaRo86], a sequence of packets that have the same source and destination addresses and ports, is a well known reason for the lack of independence of interarrival times. The authors also performed further analysis to check for independence if packet trains are removed. Their experiments show that even without packet trains interarrival times are not independent. This led to the conclusion that packet trains are not the only reason for dependencies in interarrival times.

The authors in **[HoVe03]** aim at the recovery of the ***spectral density*** of the packet arrival process and the distribution of ***number of packets per flow*** from sampled values. They are only interested in the packet arrival process and do not consider packet sizes. They investigate what information can be inferred from packet sampling and flow sampling. In both cases probabilistic sampling is used, where each element is selected independently with a probability p. They show by inversion techniques that it is possible in theory to recover the

packet size distribution and the spectral density from packet level sampling. Nevertheless, the practical estimation quality is only poor for small selection probabilities ($p<<0.5$). With flow sampling much better practical results could be achieved for both investigated metrics.

3.6.4 Sampling Synchronization for Multipoint Measurements

In **[CoGi98]** sampling techniques for the estimation of QoS parameters in ATM networks are investigated. The authors investigate a content-based selection scheme, in order to synchronize the selection of ATM cells at multiple points. A random reference pattern is compared with parts of content of an arriving ATM cell.

A hash-value is generated for the selected cells, which is used to recognize the cell at different observation points (like a packet ID, see section 2.4.4.4). This hash value is reported together with the timestamp of the arrival time to an analysis component. The measurement results are used to estimate the cell loss ratio (CLR) and the mean cell transfer delay (CTD). The authors compare the achieved estimation accuracy of this content-based scheme with the expected accuracy for simple random sampling. The accuracy is expressed by the variance of the estimates. Results from simulations and with a real ATM traffic trace show that the accuracy is very close to the expected accuracy. From this the authors conclude that they can realize a random selection by using the proposed content-based selection method.

In **[DuGr00]** Nick Duffield and Matthias Grossglauser propose an approach, called ***trajectory sampling***, for calculating the paths (trajectories) that packets take through the network based on samples taken at different observation points. The idea is further elaborated in [DuGr01], [DuGG02], [DuGr03] and [DuGr04]. A hash-based packet selection is used to synchronize selection processes at different observation points. The packet selection is based on a hash function on invariant packet header fields and parts of the content. A similar hash function is used to calculate a packet ID for the correlation of packet arrivals at different measurement points. A problem with hash-based packet selection is that it is a deterministic function on the packet content[6]. Therefore the packet selection is biased, that means that packets with specific content are more likely to be selected than others. For the estimation of the packet trajectories it is important to avoid bias towards specific addresses. Therefore the statistical properties of the hash function are evaluated with empirical investigation on four packet traces, each with 1 million packets. The traces were measured on a LAN segment close to the border of a campus network. For these traces the distributions of address prefixes in the population and in the sample are compared using the Chi-square test. If 40 bytes are used as input for the hash function, the distributions look quite similar, that means the samples seem to be selected independently from the address prefixes. Nevertheless, for hash-based selection methods in

[6] In the PSAMP terminology a deterministic selection based on content is called a filter. Therefore hash-based sampling is categorized as a filtering technique.

general it would be interesting to investigate whether this assumption also holds for other traces and other packet attributes. Further investigations on the suitability of different hash functions for the emulation of random selection are in progress (**[NiMD04]**, **[NiMT04]**).

3.6.5 Implementations

Some measurement tools already include sampling functions. Furthermore, prototypes for most of the presented sampling methods exist. In this section I describe tools that are widely deployed and provide sampling support.

3.6.5.1 Network Traffic Meter (NeTraMet)

The Network Traffic Meter ***NeTraMet*** [RFC2123] is an open-source meter, which conforms to the standards of the Real-time Traffic Flow Measurement (RTFM) group, a former IETF group that now has integrated their concepts into the IPFIX working group. The meter was developed for usage-based per flow accounting and collects packet and byte counts per flow. It provides a very flexible description of classification rules, which allows a wide range of flow definition. The configuration of the meter and the data collection is done by SNMP. The NeTraMet Manager/Collector (NeMaC), which integrates configuration and collection functions, is used to download classification rules to a meter Management Information Base (MIB) and to pull measurement results from the meter. Nowadays further metrics are supported by the meter. A bi-directional packet pair matching technique was included that allows the passive measurement of round-trip times (RTT). Nevertheless, the main application for the meter is still usage-based accounting. NeTraMet now also provides a systematic sampling method that allows measuring only every nth packet instead of all packets ([Brow99] [RFC2721]). The sampling rate can be configured at startup.

3.6.5.2 Cisco NetFlow

Cisco NetFlow [NetFlow] allows passive flow measurements directly on routers. It provides useful information for a wide variety of applications like traffic engineering, traffic profiling, usage-based accounting and attack detection. Since it is integrated in Cisco routers, it is one of the most widely deployed and used measurement solutions today [Hust03]. Cisco NetFlow collects and exports per flow information that is stored in a so-called flow cache. The flow cache maintains flow entries that store a flow ID, number and bytes per flow, the timestamp of the first and last packet observed for this flow and other flow characteristics.

Flow keys are the packet attributes that are used to distinguish flows. Currently flows are specified by the following seven fields: source and destination IP address, protocol, source and destination port numbers, type of service and the (logical) input interface. A subsequent aggregation of these flows into more coarse grained flows (e.g. according only to source and destination address) can be done on the router.

Flows are considered to be unidirectional, i.e., forward and return path of a data communication are covered by different flow entries. The flow keys allow a wide variety of flow analysis. One can generate traffic matrices if flows are distinguished by source and destination. A rough analysis of application in use is possible if port numbers are investigated. The flow keys form the flow ID that uniquely identifies a flow. Currently the flow key definition is static.

Cisco uses different flow termination criteria to decide whether a flow has ended or not. A flow is considered as over if no packet has been observed for the corresponding flow ID for a configurable maximum time (default 15 seconds) or if a TCP FIN or RST packet for the flow is observed. Flow entries are also removed if they have remained for a certain amount of time in the flow cache (default 30 minutes). If the flow cache is full, the oldest flow entry is removed.

Currently flow records are exported by an unreliable UDP transmission. NetFlow version 9 [RFC3954] has been chosen to become the basis for the development of the ***IPFIX*** protocol. Cisco will provide full IPFIX support in the near future. This means for instance that the unreliable transport by UDP can be substituted by a reliable transport by the Stream Control Transmission Protocol with Partial Reliability (SCTP-PR) [RFC3758].

Several papers have addressed problems with Cisco NetFlow. NetFlow has been mainly criticized for the following reasons (e.g., [DuLu03], [EsKM04]):

- The unreliable flow record export leads to flow record loss.
- Flow termination criteria lead to unpredictable and variable reporting times.
- The enabling of NetFlow on high speed interfaces slows down router operations.

A high amount of small flows leads to the export of many flow records. Nowadays DDoS attacks or port scans generate a large number of flows that consist only of 1 packet. This generates of a huge amount of flow entries and leads to the export of an immense number of flow records. Due to the unreliable UDP transmission, flow records are discarded on the way to the collector if the traffic between exporter and collector is too high. [FeGl00] observed a flow record loss of up to 90%. Furthermore, the loss of flow records is an unintentional flow selection and can produce more severe estimation errors than the intentional packet sampling [DuLu03]. To address this problem Cisco introduced the possibility to aggregate flows before exporting them. Fixed aggregation schemes were introduced in IOS 12.0. Furthermore, in the process of moving towards IPFIX conformance Cisco will support data export by SCTP-PR soon. The use of TCP for flow record export has been considered as difficult ([Sada01], [BrCl03]). Further proposals to control the resource consumption of NetFlow and with this avoid too extensive discarding of flow records are flow-based or adaptive sampling methods as proposed in [EsVa02a] and [EsKM04].

The second problem originates from flow termination criteria that Cisco is using. Because of those different criteria flow reports are sent when different events occur (e.g., observation of

TCP-FIN, flow cache overload, etc.). For some applications it is of advantage if flow records are received at regular time intervals. [EsKM04] proposes to use fixed time bins for the flow record reporting. In [DuLT02] side effects on flow reporting are investigated that occur if sampling is enabled.

The third problem is the exhaustion of resources on the router itself due to operation of NetFlow. If processor and memory cannot keep up with high packet rates, a reduction of the routing performance cannot be excluded. This problem is addressed by Cisco by the introduction of packet sampling to NetFlow [RandNF].

NetFlow currently supports count-based systematic and stratified count-based random sampling on high end routers. Random sampling is nowadays supported on almost all Cisco hardware, except the Cisco 12000, which currently only supports systematic sampling. The sampling is performed before the classification to unburden the classification process. This makes it more difficult to provide estimates for per flow statistics, because the sampling process runs on a different data aggregation level than the analysis. Furthermore, not the sampled packets themselves but only aggregated statistics (number of packets and bytes) of the sampled packets are stored.

Cisco uses a stratified random sampling method, which is called 1-in-K sampling in this document (Figure 3-6). The measurement interval is defined by the number of packets N. The sampling method is configured by a parameter K. This parameter defines the length of a sub-interval in the measurement interval in number of packets. That means packets are stratified in accordance to the sequence in which they arrive at the observation point. A random number is taken from a previously generated table of random numbers, uniformly distributed in the range $[1, K]$. The packet that has the packet position in the subinterval that equals the random number is selected. This is repeated for every sub-interval in the measurement interval. For each subinterval a new random number is taken from the table. The set of all packets from all sub-intervals forms the sample for the measurement interval and provides the basis for the estimation.

After this the packets are classified in accordance to the flow keys and flow entries are created which contain the cumulated number of packets and the number of bytes from all sampled packets from the specific flow. The individual characteristics per packet (bytes x_i of packet i) are not known any longer at this stage. So the estimate of the flow volume and the accuracy statement for the estimate can be based on these collected statistics in the flow entries, only. Optionally flows can be aggregated afterwards by applying some pre-defined aggregation schemes.

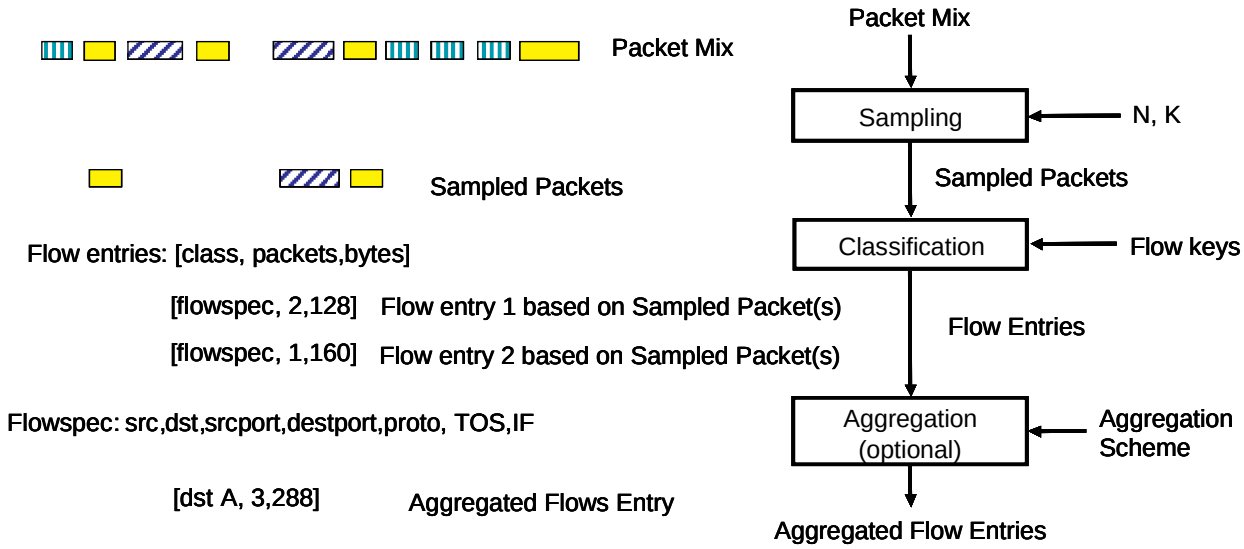


Figure 3-6: Cisco NetFlow Operation

Cisco recommends to switch to sampling for link rates of OC3 and higher. [NFperf02] and [NFperf04] contain a performance analysis of sampled and un-sampled NetFlow with respect to required resources. It is shown that a significant CPU load reduction can be achieved by enabling sampling in NetFlow compared to unsampled NetFlow.

3.6.5.3 InMon sFlow

InMons **sFlow** provides a traffic monitoring system that is used for accounting and DoS detection and explicitly supports sampling [SFLOW]. In contrast to NetFlow, which only provides per flow information, sFlow allows the export of per packet information (e.g., packet headers). The data records (sFlow datagrams) are sent via UDP to collectors. Sampling methods are implemented in measurement processes (sFlow agents) embedded in routers/switches or network probes. [RFC3176] describes the sampling methods and the reporting format used by the sFlow traffic monitoring system. sFlow uses two mechanisms: packet-based sampling of packets in switches/routers and time-based sampling of network interface statistics. After the selection the packet header or packet features are copied or extracted from sampled packets.

The packet-based sampling randomly selects packets from the flow of packets that are received on one interface of the device and are forwarded to another interface[7]. To realize the random selection a skip-counter is set to a random number and decremented each time a packet arrives. If the counter is reduced to zero the arriving packet is sampled. Then the skip-counter is set again to a random integer. Three counters are maintained: the number of all

[7] The term flow here is used to describe all packet that flow from one interface to another. This is no contradiction to the general flow definition used here. Nevertheless, NetFlow usually uses more fine granular flows (defined by the flow keys).

packets seen in the measurement interval (parent population), the number of sampled packets (sample size) and the skip-counter. It is possible to have one sample entity per IF with separate state and parameters.

With the time-based sampling of network interface statistics, interface counters are polled by sFlow agents. Samples of the SNMP counters are piggybacked to the packet samples. Configuration of sFlow is done by SNMP with the sFlow MIB or by command line. It is possible to configure among other parameters (like sFlow data source, collector address, etc.) also the sampling rate and sampling interval.

In contrast to NetFlow, which aggregates per packet information into a flow record, sFlow reports directly the per packet information (e.g., packet headers) of all sampled packets. That means with sFlow one has more information available after the sampling process. On the other hand capturing and result transport requires more resources than flow export with NetFlow.

3.6.6 Summary

As described in the previous chapters, recently much work was published on the estimation of packet counts and traffic volume. Those metrics are of main interest for usage-based accounting. Table 3-2 summarizes existing work on the estimation on packet count and volume estimation. Those references are useful for the selection of schemes for usage-based accounting and will be further evaluated in chapter 4.3.

Reference	**Metric**	**Schemes**	**Aggregation Level**	**Method**
[AmCa89]	peak load, packet count	Time-based and combined systematic and random	All packets on link, no classification	Only general rules
[JePP92]	Packet count	count-based probabilistic	Sampling before Classification (src address)	Theory only
[ClPB93]	Distribution of packet sizes	Count- and time-based, systematic, random and stratified	All packets on link, no classification	Empirical only
[DrCh98]	packet count mean and variance	Adaptive systematic count-based	Sampling before classification	Theoretical and empirical
[DuLT01]	Packet count, volume	Size-dependent flow sampling	Sampling (of flows) after classification	Theoretical and empirical
[EsVa01, EsVa02a-b, EsVa03]	Packet count, volume	Non-uniform (content-based) probabilistic	Sampling after Classification	Theoretical and empirical
[DuLT02]	Packet count, volume	count-based probabilistic	Sampling before classification	Theoretical and empirical
[ChPZ02a-c]	Volume (traffic load)	Adaptive count-based probabilistic	All packets on link, no classification	Theoretical and empirical

Reference	Metric	Schemes	Aggregation Level	Method
[DuLu03]	Packet count, volume	count-based probabilistic	Sampling before classification	Theoretical and empirical
[HoVe03]	Distribution of number of packets per flow	probabilistic (packet and flow sampling)	(Sampling before classification) [8]	Theoretical and empirical
[KoLM04]	packet rate	Flow-state-based filtering	Sampling after classification	Theoretical and empirical
[KuXW04]	packet count	lossy data structure, probabilistic	Sampling before (probabilistic) and after classification (lossy data structure)	Theoretical and empirical
[EsKM04]	Packet count, volume	Adaptive probabilistic	Sampling before classification	Theoretical and empirical

Table 3-2: Existing Work for Packet Count and Volume Estimation

Sampling should be applied as early as possible in sequence of operations in the measurement process in order to achieve a maximum resource reduction by unburdening subsequence processes. Therefore the most interesting approaches are those that work with sampling before classification. The existing approaches that work with this aggregation level apply either systematic or probabilistic sampling.

There are much fewer publications that are useful for the deployment of sampling for SLA validation. SLA validation requires QoS measurements. These days QoS measurements are often done with active measurements. Therefore there are only few publications on the use of sampling methods for QoS measurements.

The authors of [CoGi98] show how the cell loss ratio (CLR) and the mean cell transfer delay (CTD) for an ATM network can be estimated from samples. They apply statistical methods for random sampling to a content-based selection and show that with the content-based method a similar accuracy as expected for random sampling can be achieved. Also in [DuGr04] it is described how the trajectory sampling approach, which is based on a content-based packet selection, provides a solution for sampling synchronization in multipoint measurements. This is an important feature for QoS measurements. In [NiMD04] the suitability of a hash-based packet selection for the measurement of one-way delay is mentioned.

3.7 Research Plan

This chapter points out the research plan and the methodology that is applied. More details on selection criteria, assessment of existing work and the selection of appropriate schemes for the target scenarios can be found in chapter 4 and 6.

[8] Not explicitly stated in paper.

3.7.1 Scope of this Work

Goal of this work is the selection and evaluation of applicable sampling schemes for key applications. Subtasks are to investigate how and to what degree statements about the estimation accuracy can be made, how the accuracy can be approximated during the operation and whether modifications of schemes allow an increase of the estimation accuracy. This all is needed to show whether and how providers can substitute full measurements by sampling.

Figure 3-7 shows the relations between customer traffic and measurement demands. Customer applications and customer demands determine what quality demands should be met and with this what metrics are of interest. The customer applications also form the flows that enter the network and with flows from other customers form the traffic within the network.

From the metrics of interest one can derive what measurement methods are required. With this one can investigate what sampling schemes might be applicable. For the schemes one can attempt to derive a model to express the expected accuracy. The accuracy can depend on the statistical properties of the population (here the traffic mix in the network). The investigation of the relations between metrics, schemes, model and statistical properties of the traffic is the focus of this work.

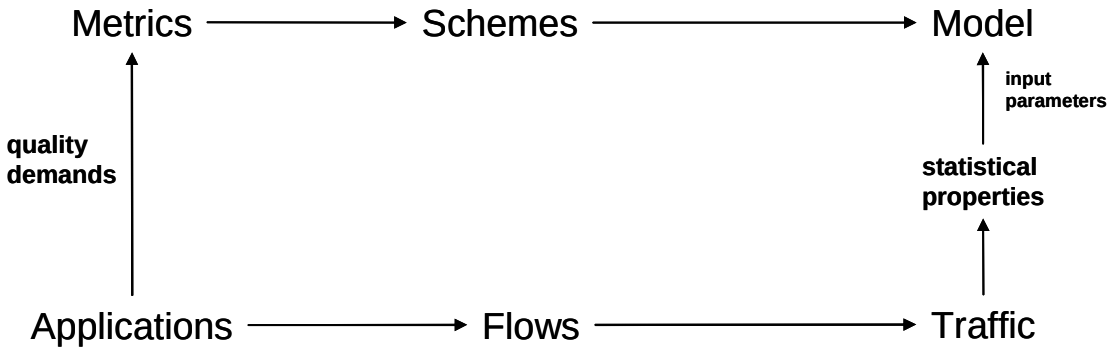


Figure 3-7: Relations

Figure 3-8 shows more precisely which relations are investigated in this work. For the most promising schemes it is investigated if and how the expected accuracy can be expressed in a model. It is analyzed which traffic characteristics influence the accuracy and how those characteristics can be calculated, approximated or predicted to provide an accuracy statement during operation. For this any a-priori knowledge about the customer's flows can provide valuable information. The relations between applications, flows and the traffic mix in the network depend on the number and type of flows from different customers, on scheduling and queuing methods in the network. Investigating those relations is a huge subject of current research but it is not target of this work.

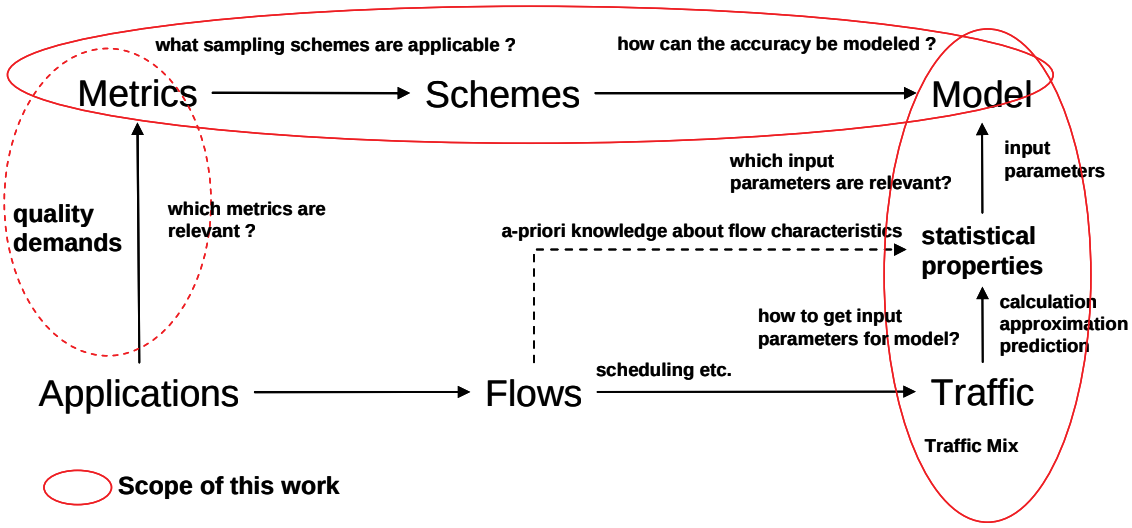


Figure 3-8: Scope of Work

In this work I constrain myself to sampling methods for ***non-intrusive*** measurement methods. In order to get the highest possible resource reduction, the data selection should be done as early in the measurement process as possible. Therefore I concentrate on ***packet selection*** methods in contrast to flow selection, which can only be applied after packets have been classified.

First a general ***taxonomy*** is developed for definition and differentiation of packet selection methods for IP measurements. Parts of this taxonomy have been contributed to standardization in the IETF PSAMP group [ZsMD05].

For the investigation of packet selection schemes I focus on two example applications, for which the need for sampling deployment is crucial (see chapter 2.5): ***usage-based accounting*** and ***SLA validation***. Mathematical modeling of a sampling scheme depends on the investigated metric and on the available information in the specific scenario. Therefore different models need to be developed for the same scheme to model the estimation accuracy for different metrics and different scenarios. Since an in-depth investigation of all available schemes would go beyond the scope of this work, the effort is reduced by ***pre-selecting*** the most promising schemes for the selected metrics. Requirements and criteria are defined for the usage of sampling for the selected scenarios. Based on those criteria a selection of appropriate selection methods is done for an in depth investigation and comparison. Since requirements for the selected scenarios differ, different schemes may be applicable for usage-based accounting than for SLA validation.

Furthermore, I especially investigate the use of ***stratified sampling*** schemes for both scenarios because I am convinced that such methods have a high potential to improve sampling performance and little has been done so far to explore the use of such methods. The pre-selection of schemes for usage-based accounting is described in chapter 4. The pre-selection of schemes for SLA validation is described in chapter 6.1. The selected methods are then investigated by theoretical modeling and empirical experiments.

The incorporation of practical aspects from a real router vendor perspective was possible with the project VEGAS, funded by Cisco Systems. I had the opportunity to work directly together with Cisco engineers and to get very detailed information about Cisco NetFlow operations. Based on the mathematical modeling and empirical comparison of schemes in this work I could discuss improvements and recommendations directly with Cisco engineers.

3.7.2 Positioning in the Research Field

Figure 3-9 shows the most important milestones in the research field on sampling for IP measurements for the metrics that are of interest for the target applications (accounting and SLA validation). In 1989 the first approach started to estimate the overall packet count on a link from sampled packets. Since then, the packet count on a link became a less interesting metric because packet counters became widely available. Instead, the packet count per flow became of interest.

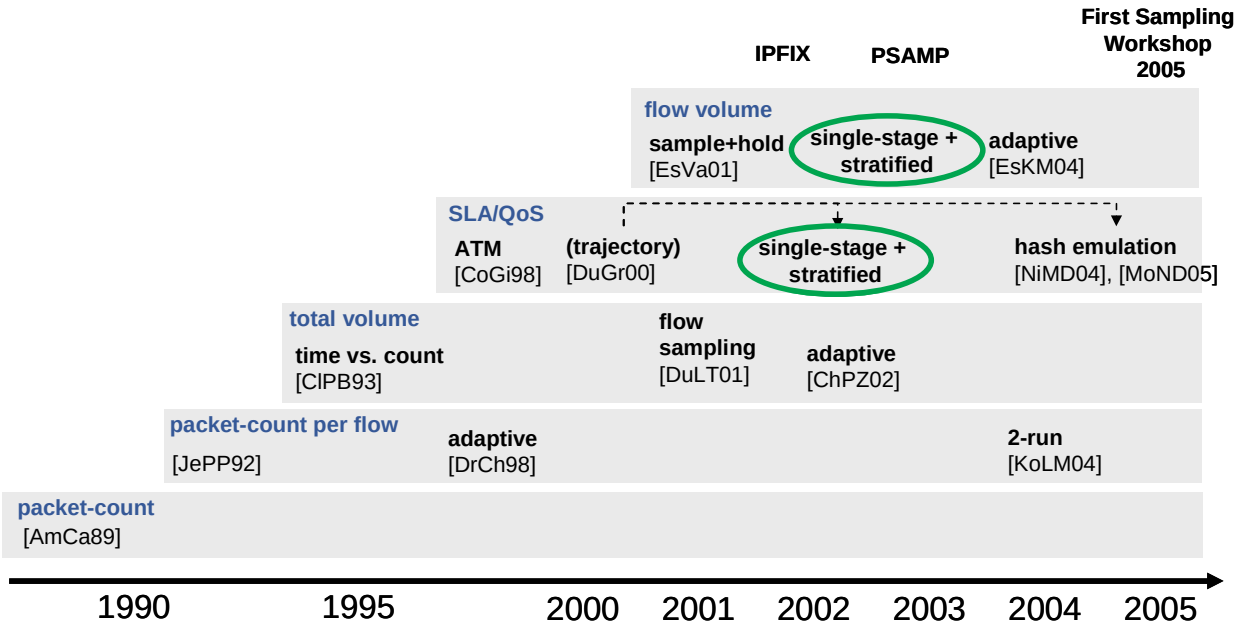


Figure 3-9: State of Art Overview

In 1992 the packet count per flow became of interest In [JePP92] the packet count from a specific source is estimated for accounting purposes. In [DrCh98] a first adaptive approach is presented, which adapts sampling parameters to available CPU power. In [KoLM04] an new approach for estimating the packet count per flow is presented which bases on the estimation of 2-runs (two consecutive packet of a flow). With this a bias towards larger flows is introduced. In 1993 different time- and count-based schemes were compared empirically for the estimation of the total volume. Later in 2001 an approach for flow sampling was introduced for the estimation of the volume. In 2002 an adaptive approach for estimating the volume was presented with the goal to control the accuracy.

In 2001 approaches to estimate the flow volume evolved mainly due to the need of usage-based accounting [EsVa01]. The sample&hold approach realizes a non-uniform probabilistic

sampling that biases the selection already towards large flows. A disadvantage of the sample&hold approach is the need to classify all packets (i.e. sampling is done after classification). In 2004 an adaptive approach for flow volume estimation was proposed to control the resource consumption of the measurement functions [EsKM04]. The first part of my work is focused on the flow volume estimation with different single-stage methods (see chapter 4.1) and with a stratified approach (green circle).

There are only few publications related to QoS measurements and SLA validation. In 1998 an approach was published to measure cell loss ratio (CLR) and cell transmission delay (CTD) in ATM networks. In 2000 an approach for measuring trajectories (paths that packets take in a network) was published in [DuGr00]. This approach is not related to SLA validation but describes the synchronized selection of packets at different observation points. It shows how a random selection can be emulated by such a hash-based selection. The work is further elaborated in [DuGr01], [DuGG02], [DuGr03], [DuGr04]. In my work I propose to use this for multipoint synchronization for multipoint QoS measurements (e.g. one-way-delay) [Zseb02]. Apart from different single-stage methods (see chapter 6.1.) I investigate the applicability of stratified sampling. Nowadays further work has been started to evaluate the quality of the emulation of random sampling for different hash-functions ([NiMD04], [MoND05]).

Table 3-3 shows an overview of existing publications and the positioning of this work with regard to metrics (relevant for the target applications) and selection methods. Some publications contain only empirical investigations. Those are denoted with an (E). Publications that contain only theoretical work are denoted by a (T). The metrics and schemes that are investigated in this work are marked by “This work” and a green field color. The different single-stage sampling schemes considered in this work are further assessed and selected in the section on the target applications (chapter 4.1 and 6.1).

Metric	Single-Stage	Adaptive (to control resources)	Adaptive (to control accuracy)	Stratified	Flow sampling (only)	Flow sampling (combined)	Others
Packet count (link)	[AmCa89] (T)				[DuLT01] [DuLT04] [DuLT05a]	[DuLu03]	
Packet count per flow	[JePP92] (T) [DuLT02] [HoVe03]	[DrCh98] [EsKM04]			[HoVe03] [DuLT01]	[EsVa01] [EsVa02a-b] [EsVa03]	[KoLM04] (2-run) [KuXW04] (sampling + lossy data structure)
Total volume	[ClPB93] (E)		[ChPZ02a-c]	[ClPB93] (E)	[DuLT01] [DuLT04] [DuLT05a]	[DuLu03]	
Flow volume	This work [DuLT02] [DuLu03]		[EsKM04]	This work	[DuLT01]	[EsVa01] [EsVa02a-b] [EsVa03] [DuLu03]	
QoS metrics	This work [CoGi98]			This work			
Other metrics	[DuGr00] (trajectory) [MoUK04] (large flows) [DuLT02] (side effects) [ClPB93] (E) (inter-arrival times)	[DrCh98] (hurst parameter)		[ClPB93] (E) (inter-arrival times)	[HoVe03] (spectral density)		[PASF02] (temporal correlations)

Table 3-3: State of Art and Positioning of this Work

3.7.3 Methodology

The taxonomy is defined based on existing work in research and in standardization. The developed categorization is used to clearly distinguish between different sampling methods. Requirements and criteria are derived from the measurement methods and metrics that are needed in the selected scenarios under consideration of potential limitations that may apply. The selection of methods is done based on those criteria. It is analyzed what methods are already covered by existing work and where theoretical or practical investigations are lacking.

Figure 3-10 shows the methodology used. First the two target applications, the relevant metrics and appropriate measurement methods are selected. In accordance to requirements that originate from those scenarios applicable schemes are chosen with regard to the expected benefit. Those schemes are further investigated by mathematical modeling and empirical investigations. From traffic characteristics of real traffic traces the practical achievable accuracy can be derived and it can be investigated how the accuracy evolves in cases where model assumptions do not hold. Furthermore one can derive further ideas for accuracy improvement from information about the traffic. One example is the selection of suitable

variables for the grouping of elements for stratified sampling techniques. From the dynamics of the traffic characteristics the predictability of the accuracy estimation can be assessed.

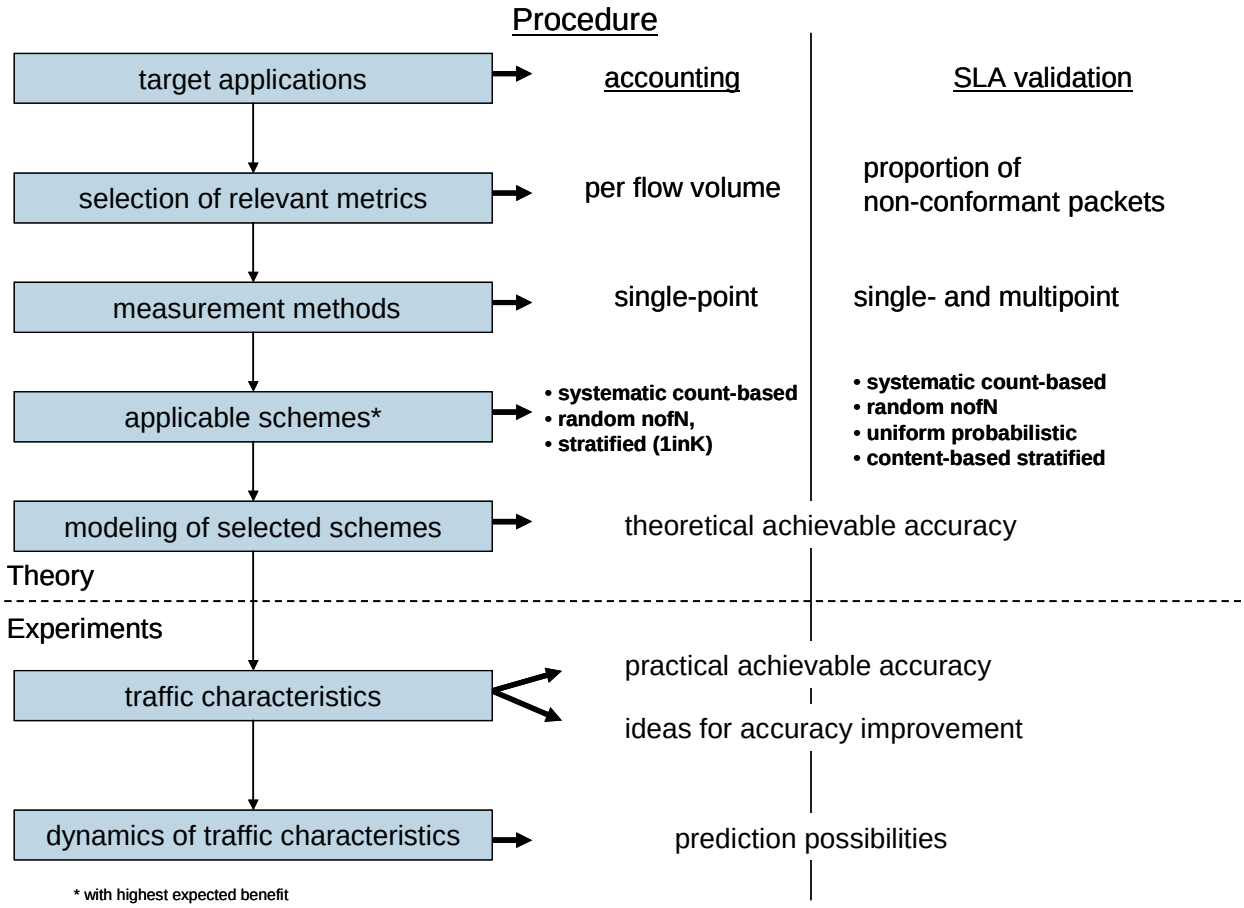


Figure 3-10: Methodology

The assessment of the quality of the estimation is done as follows. In reality the subset of elements (the sample) is selected only once and one gets exactly one estimate for the metric of interest. If the real value from the population is known one can calculate the estimation error. With this one can assess the accuracy of this single sample run. But it provides no information about the accuracy of the scheme in general, because the estimation error can be totally different for another sample run. Therefore instead of considering a single sample run, one considers how the estimation error would evolve if infinite sample runs were performed. This is expressed by the distribution of the estimate (Figure 3-10). Based on these considerations the quality of a sampling scheme is assessed by the two following criteria: bias and precision.

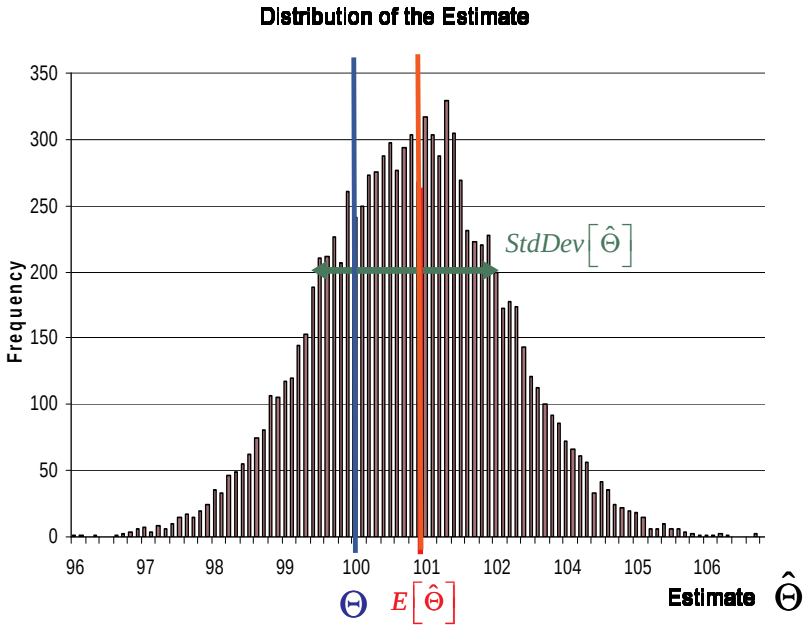


Figure 3-11: Distribution of the Estimate

The ***bias*** quantifies how far the estimates from all sampling runs lie from the exact value. The distance is measured by the difference of the ***expectation*** of the estimate $\hat{\Theta}$ to the real value.

$$Bias = E\left[\hat{\Theta}\right] - \Theta \tag{3.4}$$

If the expectation of $\hat{\Theta}$ equals the real value Θ the estimate is unbiased.

$$\text{Estimate is unbiased if:} \qquad E\left[\hat{\Theta}\right] = \Theta \tag{3.5}$$

The relative bias is used to compare results from experiments were the real value differs (e.g. bias for different flows).

$$Bias_{rel} = \frac{E\left[\hat{\Theta}\right] - \Theta}{\Theta} \tag{3.6}$$

The ***precision*** quantifies how much the estimates from multiple runs scatter around the mean. The ***variance*** is used to asses the precision of an estimate. A small variance of estimates corresponds to a higher precision.

Precision of estimate $\hat{\Theta}_2$ is better than for estimate $\hat{\Theta}_1$ if:

$$V\left[\hat{\Theta}_2\right] < V\left[\hat{\Theta}_1\right] \tag{3.7}$$

So a key metric to expresses the achievable accuracy of a sampling scheme is the expectation and the expected variance of the estimate. The standard deviation of the estimate (also called standard error) can be calculated easily from the variance. The advantage of using the standard error is that the units are the same as for the estimate and the results are more understandable. The ***absolute standard error*** shows the absolute deviation of the estimate $\hat{\Theta}$ and therefore can be used as a measure of the precision.

$$StdErr_{abs} = \sqrt{V\left[\hat{\Theta}\right]} \tag{3.8}$$

The ***relative standard error*** (or coefficient of variation) shows the relative deviation as percentage of the real value. It allows a better comparison of results and eliminates effects that are due to changes of the value itself (e.g., higher absolute error for higher values). For this the error is divided by the real value:

$$StdErr_{rel} = \frac{\sqrt{V\left[\hat{\Theta}\right]}}{\Theta} \tag{3.9}$$

These metrics are used throughout this work to asses and compare the performance of different sampling schemes.

The standard error is also used to define a ***confidence interval***. A confidence interval gives the probability that the real value lies within given boundaries. The confidence boundaries and confidence level show in an illustrative way what accuracy can be expected. Therefore those values are useful in order to communicate the expected accuracy to customers.

If it is assumed that the estimate is normal distributed[9], the confidence interval can be defined as follows:

$$Prob\left(\hat{\Theta} - z_c \cdot StdErr_{abs}\left[\hat{\Theta}\right] \le \Theta \le \hat{\Theta} + z_c \cdot StdErr_{abs}\left[\hat{\Theta}\right]\right) = 1-\alpha \tag{3.10}$$

That means that the probability Pr that the real value Θ lies within the given confidence boundaries is *1-α*. 1-α is called the confidence level. z_c is the percentile of the normal distribution for *1- α/2* and is called the critical value. The critical value is derived from the normal distribution for a specific confidence level. So if a higher confidence level is desired for the same estimate with the same sampling parameters, the confidence interval will get larger. One can deduce that with a probability of *1-α* the estimation error $\varepsilon = \hat{\Theta} - \Theta$ is not larger than the standard error multiplied by the critical value.

$$Prob\left(\varepsilon \le \left| z_c \cdot StdErr_{abs}\left[\hat{\Theta}\right]\right|\right) = 1-\alpha \tag{3.11}$$

For a level of confidence of *1-α =0.683* one gets a critical value of $z_c=1$. That means the standard error provides the confidence boundaries for a confidence level of 68.3. That means only with a probability of 68.3 % the estimation error ε is not larger than $\left| z_c \cdot StdErr_{abs}\left[\hat{\Theta}\right]\right|$.

The critical values for the more common confidence levels of 95% or 99% are $z_c=1.96$ and $z_c=2.58$, respectively. When looking at the standard error for the comparison of schemes, one need to keep in mind that the standard error has to be multiplied by a factor >1 to get the more often requested confidence levels of 95% or 99%.

[9] In most cases one can assume that the estimate is at least asymptotically normal distributed. If a normal distribution cannot be assumed one needs to substitute the critical value z_c by another critical value derived from the real distribution of the estimate.

Mathematical models are developed to express the bias and expected accuracy for a given sampling scheme. They show how bias and precision depend on traffic characteristics and sampling parameters. In some cases it is necessary to make initial assumptions in order to derive a model. Those assumptions are scenario and scheme-specific and are described in chapter4 and 6.

Empirical investigations are used to check whether assumptions, that where made for the modeling, hold true in reality. For scenarios where the compliance with assumptions cannot be guaranteed it is investigated to what degree empirical results differ from the model.

For the empirical investigation traffic traces available from other measurement groups are used. In addition to this own measurements are performed. A full (offline) analysis of the traces is performed to get the real values for the metric of interest (as reference) and to gain knowledge about trace characteristics (number and size of flows, packet size distribution, correlations, etc.). Traces are split into measurement intervals, because for real-time measurements estimates will be updated regularly and therefore have to be calculated per interval.

The empirical bias and standard error of the estimates is investigated for the selected scheme with different parameter settings. In order to capture the random variation of the estimate a high number of sampling runs is simulated with the same sampling parameter setting over the same measurement interval. For each run an estimate for the metric of interest is calculated. All runs are repeated for all measurement intervals in all available traces. The result provides the empirical variation of the estimate for one specific parameter set. That means the procedure has to be repeated for each investigated scheme and parameter setting.

The empirical bias of the estimates is calculated as the difference between the mean value of the estimates of all *R* sampling runs and the real value.

$$Bias_{abs,emp} = \left(\frac{1}{R} \cdot \sum_{r=1}^{R} \hat{\Theta}_r \right) - \Theta \tag{3.12}$$

The relative empirical bias is the absolute empirical bias divided by the real value.

$$Bias_{rel,emp} = \frac{\left(\frac{1}{R} \cdot \sum_{r=1}^{R} \hat{\Theta}_r \right) - \Theta}{\Theta} \tag{3.13}$$

The empirical standard error is calculated as the standard deviation of the estimates from all *R* sampling runs.

$$StdErr_{abs,emp}\left[\hat{\Theta}\right] = \sqrt{V\left[\hat{\Theta}\right]} = \sqrt{\frac{1}{R} \cdot \sum_{r=1}^{R} \left(\hat{\Theta}_r - E\left[\hat{\Theta}\right]\right)^2} \tag{3.14}$$

The relative empirical standard error is the absolute empirical standard error divided by the real value.

$$StdErr_{rel,emp}\left[\hat{\Theta}\right] = \frac{\sqrt{V\left[\hat{\Theta}\right]}}{\Theta} = \frac{\sqrt{\frac{1}{R}\cdot\sum_{r=1}^{R}\left(\hat{\Theta}_r - E\left[\hat{\Theta}\right]\right)^2}}{\Theta} \tag{3.15}$$

The empirical results are compared with the theoretically expected values derived from the mathematical model. Trace analysis and sampling simulation is done by C and C++ programs developed in the VEGAS project [VEGAS-SW], statistical analysis and graphical representation of results is done with the statistic software R [R] and with Microsoft Excel.

4 Sampling Techniques for Usage-based Accounting

This chapter describes the theoretical investigation of sampling schemes for usage-based accounting. Requirements for accounting scenarios are pointed out and existing work is assessed with regard to the requirements. Based on this, schemes are selected for a further investigation. Then mathematical models are developed for the selected schemes.

4.1 Measurement of the Flow Volume

This work focuses on usage-based accounting that is based on the flow volume (see 2.5.1). The flow volume Sum_f for flow f is calculated from the packet sizes $x_{i,f}$ of all N_f packets in the flow as follows.

$$Sum_f = \sum_{i=1}^{N_f} x_{i,f} \tag{4.1}$$

The flow volume is an aggregated metric. In order to calculate the exact flow volume all incoming packets need to be classified and the packet sizes of all packets that belong to flow f have to be summed up. Usually the individual packet sizes $x_{i,f}$ are discarded. If also the number N_f of packets is stored one can calculate the mean packet size as follows.

$$\mu_f = \frac{1}{N_f} \cdot \sum_{i=1}^{N_f} x_{i,f} \tag{4.2}$$

4.2 Requirements

Calculating the flow volume requires the examination of the packet headers from all packets that belong to the flows of interest in order to evaluate the packet sizes. Packet selection schemes are used to reduce the number of packets that have to be processed. A reduction of processing resources can only be achieved if this header analysis needs to be done for fewer packets. As a consequence, all selection schemes that require packet content analysis are inappropriate for accounting scenarios. The examination would cost equal or more effort than the examination of the packet sizes themselves. Therefore an essential requirement for sampling schemes for the accounting scenario is that it works without content analysis.

A desired property for accounting would be to get a constant accuracy for the volume estimation. With a constant accuracy the magnitude of a potential monetary loss or gain would be stable for providers and customers.

Another desirable feature is the control of sampling parameters like population size and sample size. Those values are usually key parameters for an assessment of the estimation accuracy. Controllable sampling parameters allow a more accurate accuracy assessment and provide the basis for adaptive schemes. Furthermore, in order to be applicable to arbitrary traffic profiles, it would be of advantage if the estimation accuracy would be independent of correlations in packet sequence.

Some features that would be nice-to-have are the operation without link counters or random number generation, and the possibility to get reports of the results at fixed time-durations. One can summarize that the following attributes would be of advantage for a sampling method for the accounting scenario.

- Works without content analysis
- Provides constant accuracy
- Sampling parameter controllable (population size and sample size)
- Accuracy independent of correlations in packet sequence
- Works without link counters
- Works without random number generation
- Fixed time-duration per report

In the following I assess which of the possible schemes best fits the desired features.

4.3 Assessment and Selection of Schemes

In order to select suitable sampling schemes for usage-based accounting two steps are performed:

- Assessment of Basic Schemes: first it is checked which basic schemes fulfill criteria defined in 4.2. Schemes that are not applicable are ruled out and the most promising candidates are selected.
- Assessment of Existing Work: In a second step existing work in this area is investigated and it is checked what can be used from their findings.

4.3.1 Assessment of Basic Schemes

Table 4-1 shows an assessment of all basic schemes (as defined in 3.3) for usage-based accounting with regard to the requirement and criteria defined in 4.2. The notation introduced in 3.3.6 is used as scheme identifier. Mandatory requirements are marked in red, desired or recommended attributes in blue and optional or nice-to-have features in green. The last row contains references to related work where such schemes were investigated.

	T/-/RP	T/T/S	T/T/RP	T/C/S	T/C/RP	T/C/RN	T/Co/S	T/Co/RP	C/-/RP	C/T/S	C/T/RP	C/C/S	C/C/RP	C/C/RN	C/Co/S	C/Co/RP
No content analysis required	✓	✓	✓	✓	✓	✓	-	-	✓	✓	✓	✓	✓	✓	-	-
Constant accuracy	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Controllable population size	-	-[10]	-	-	-	-	-	-	✓	✓	✓	✓	✓	✓	✓	✓
Controllable sample size	-	-[11]	-	-	-	-	-	-	-	-[11]	-	✓	-	✓	-	-
Independent of correlations in packet sequence	✓	-	-	-	✓	✓	-	✓	✓	-	-[12]	-	✓	✓	-	✓
No link counters required	✓[13]	✓	✓[13]	-	-	-	✓	✓	-	-	-	-	-	-	-	-
No random number generation required	-	✓	-	✓	-	-	✓	-	-	✓	-	✓	-	-	✓	-
Fixed report times	✓	✓	✓	✓	✓	✓	✓	✓	-	-	-	-	-	-	-	-
Related Work		[AmCa98], [ClPB93]	[AmCa98]		[ChPZ02]				[DuLT02], [DuLu03]			[DuLT02], [DuLu03]	[DuLT02], [DuLu03]		[DuGr00]	

Table 4-1: Evaluation of Schemes for Usage-based Accounting

As can be seen in Table 4-1, content-based schemes are not suitable for accounting purposes. The effort of a content-based selection equals or exceeds the effort of a packet size analysis for all packets (see 4.2). None of the basic schemes provides a constant accuracy. The problem is that the accuracy for all schemes varies with population characteristics. So a constant accuracy can be only achieved with an adaptation of configuration parameters.

Schemes with count-based measurement interval definition allow a controllable size of the parent population. But only systematic and random n-out-of-N sampling allow also to control the sample size. With probabilistic sampling an independent choice is made per packet. Therefore the sample size varies. An accuracy independent of potential correlations in the

[10] Possible if time-based extrapolation is used.

[11] Sample size can be controlled if mixed triggers are used.

[12] Independent if mixed triggers are used and stop trigger set to 1 packet.

[13] But counter needed for count-based extrapolation.

traffic can only be achieved with random sampling schemes. Systematic schemes have the advantage to work without random number generation, but on the other hand do not provide an accuracy independent of potential correlations. Only schemes with time-based measurement interval definitions allow a fixed reporting time and can work without link counters.

4.3.2 Selection of Schemes

In accordance to the evaluation the following schemes are selected for an in-depth investigation for the metric volume:

C/C/S: Systematic count-based sampling can be implemented and operated with minimal effort. It requires only a packet counter and no random number generation. Due to the lightweight selection process, it can operate at high speeds. Nevertheless, because of the systematic selection process, it is likely that correlations in the population influence the estimation accuracy.

C/C/RN: n-out-of-N sampling is a random sampling scheme that can easily be implemented and requires only few resources. Due to the random selection, statistical models can be applied. Furthermore, count-based stratified schemes, as used in Cisco routers, can be derived from n-out-of-N sampling.

As mentioned above a special focus is set towards stratified schemes. Therefore additionally a stratified scheme is investigated and it is checked whether an improvement can be achieved by using stratification:

Stratified C/C/RN (1-in-K): 1-in-K sampling is a pseudo multi-stage scheme that can be considered as a combination of a classification followed by a random selection. For 1-in-K sampling the measurement interval is first split into blocks of K packets (count-based stratification). This classification of packets in accordance to the packet count can be considered as multiple systematic selections based on the packet count (C/C/S) and is called count-based stratification. After this one packet is randomly selected from each block with a count-based random n-out-of-N selection (C/C/RN). This scheme is implemented in Cisco routers.

The schemes below are ***not*** considered for an in-depth investigation due to the following reasons:

T/T/S, C/T/S: Systematic time-based sampling is simply and inexpensive to implement because no counters are required. It is suitable for time-related metrics like packet rates and load changes over time. But the scheme has quite limited applicability if one wants to estimate the transmitted overall volume or QoS parameters. With time-based measurement intervals, the population size N varies. Furthermore, the number of packets n that is selected heavily depends on the packet rate within the sample interval. So in addition to the dynamic traffic characteristics also the sample fraction (n/N), a key parameter for providing an

accuracy statement, underlies heavy variations. If one derives a statistical model for this, both, the unknown traffic characteristics and the varying sampling parameters, need to be considered as random variables. Due to the high variability of the components, a high variance of the estimates and with this only a small accuracy is expected. In addition to this, one usually has no idea about the distributions of those random variables (RVs), which makes it difficult to derive a model at all.

A further problem occurs due to the systematic selection process. Usually one cannot preclude correlations in the trace. If periodicities in the population interfere with the systematic selection process, one can get biased estimates. So the bias and estimation accuracy additionally depends on the unknown sequence at which packets are observed. Furthermore, I assume that correlations between subsequent packets are more likely than periodicities at higher lags. With systematic time-based sampling one always select a block of subsequent packets in the time interval. Therefore one gets additional bias, if there are dependencies between directly successive packets. In addition to this, packet counters are nowadays widely available and also work with high packet rates (e.g., Cisco Catalyst). So the ability to work without counters is only a small advantage compared to other schemes.

T/T/RP: Random probabilistic time-based sampling has also the problem that population and sample size varies (see T/T/S).

T/C/S, T/C/RP, T/C/RN: Those schemes work with time-based measurement interval and count-based trigger. As explained above, this can lead to some difficulties. Furthermore, the schemes require a link counter for the sampling selection anyway. If a link counter is present, it is not much effort to also use this for the interval definition. Since the time-based schemes do not provide a controllable parent population, I expect to get better results with the same schemes with a count-based measurement interval definition. Therefore I concentrate on the variants C/C/S and C/C/RN with count-based measurement intervals.

C/T/RP: This scheme works with a count-based measurement interval and a time-based trigger. As for the schemes above it requires a link counter. Due to a controllable sample size I expect better results with the count-based variants C/C/RP and C/C/RN.

T/-/RP, C/-/RP: Uniform probabilistic sampling requires nearly the same effort than n-out-of-N sampling. Nevertheless, due to the selection decision per packet the number of selected packets can differ from the target sample size. If this variability of the sampling parameter is reflected in a model, I expect a smaller theoretical accuracy than for n-out-of-N sampling. Apart from this I expect no significant differences from n-out-of-N sampling. Nevertheless, the difference between uniform probabilistic and n-out-of-N sampling will be investigated for quality conformance measurements, because an advantage for multipoint measurements is expected.

C/C/RP: Non-uniform probabilistic sampling is similar to uniform probabilistic but the selection probability can vary for each packet. With this one could realize an intentional

biased selection. This is an interesting feature, but probabilistic sampling in general has the problem of varying sample sizes which complicates the accuracy prediction. Furthermore, the required probability calculation per packet increases the effort for this scheme. Therefore these schemes will not be considered in this work. Instead I will look at stratified schemes that combine classification with n-out-of-N sampling. With this a biased selection can be realized, but with a controllable sample size. In case stratification methods lead to high benefits (accuracy improvement or sample size reduction) one could consider further investigations on the differences between stratification with n-out-of-N and probabilistic sampling.

T/Co/S, T/Co/RP, C/Co/S, C/Co/RP: Filtering and ***content-based random probabilistic sampling*** requires the processing of the packet content of each packet for the selection process. The exact measurement of the volume, i.e., the analysis of the packet size of each packet, would require fewer resources. Therefore filtering-based schemes would not lead to any savings for the volume estimation.

4.3.3 Evaluation of Existing Work

In this chapter I investigate to what extend existing work can be used as basis for my investigations. There are several papers that deal with the estimation of packet count and traffic volume. Since I want to do per flow accounting, I need solutions for the estimation of the ***volume per flow***. Since packet sizes can vary extremely it is not sufficient to base accounting on packet count only. Since a lot different tariff models are out there I want to keep the flow definition as flexible as possible.

The measurement process should be unburdened as much as possible. Especially the load on packet classifiers, which cannot keep up with high packet rates, should be decreased. Therefore I am looking for solutions where packet sampling is done ***before classification*** or any other post processing.

In accordance to the evaluation of basic schemes above, I concentrate my efforts on the basic schemes ***count-based systematic*** and ***n-out-of-N sampling*** because they possess most of the important desired features (see 4.3.1). Furthermore, I want to investigate whether the use of a ***stratified scheme*** can provide an accuracy improvement

Table 4-2 shows existing work on volume and packet count estimation. It points out which publications have addressed issues relevant for our research. The table was derived from the overview table (Table 4-1) in chapter 3.6.6.

Reference	Volume Estimation	Per Flow	Systematic	n-out-of-N	Stratified	Sampling before classification ?	Theoretical	Empirical
[AmCa89]	✓	-	✓	✓	-	-	✓	-
[JePP92]	-	✓	-	-	-	✓	✓	-
[ClPB93]	✓	-	✓	✓	✓	-	-	✓
[DrCh98]	-	✓	✓	-	-	**(✓)**	✓	✓
[DuLT01]	✓	✓	-	-	.	-	✓	✓
[EsVa01,EsVa02a-b, EsVa03]	✓	✓	-	-	-	-	✓	✓
[DuLT02]	✓	✓	(-)[14]	-	-	✓	✓	✓
[ChPZ02a-c]	✓	-	-	-	.	-	✓	✓
[DuLu03]	✓	✓	(-)[14]	-	-	✓	✓	✓
[HoVe03]	-	✓	-	-	-	✓	✓	✓
[KoLM04]	-	✓	-	-	-	-	✓	✓
[KuXW04]	-	✓	-	-	-	**(✓)**	✓	✓
[EsKM04]	✓	✓	-	-	-	✓	✓	✓
Cisco NetFlow	✓	✓	✓	-	✓	✓	-	-
This work	✓	✓	✓	✓	✓	✓	✓	✓

Table 4-2: Evaluation of Existing Work

Nick Duffield ([DuLT02] [DuLu03]) and Christian Estan ([EsVa01, EsVa02a-b, EsVa03]) have considered the estimation of the volume per flow. In his publications Nick Duffield derives a model for probabilistic sampling. He argues that this model can be used also for systematic sampling because the superposition of multiple flows will lead to a randomized packet sequence. Since this assumption has not been proved and is most likely not applicable for arbitrary traces I consider the model as not necessarily as applicable for systematic sampling in general.

[14] Paper claims that results for random sampling also applicable for systematic sampling due to randomness introduced from mixing of different flows. Nevertheless this statement has not been proven.

Christian Estan also works with probabilistic sampling. In his publications he considers the volume per flow but performs the sampling after the classification. With this, classifiers still have to operate at link speed. My goal is it to apply sampling much earlier and therefore unburden the classifier. Due to the same reason Cisco NetFlow applies sampling before classification (see table). Furthermore, Cisco NetFlow implements a stratified sampling method (1-in-K) not covered so far. Stratified sampling has been used in [ClPB93] for empirical investigations on the estimation of the packet size distribution of the overall traffic mix. From this, one can derive the overall volume, but not the volume per flow. Furthermore the scheme was only empirically investigated with only one trace, no model has been provided. Since the accuracy of stratified schemes heavily depends on correlations between stratification variable and survey variable, results cannot be generalized to arbitrary traces. In my work I look at systematic, n-out-of-N and the stratified sampling method implemented by Cisco NetFlow (1inK). For those methods a high benefit is expected (see section 4.3.2) and they have not been covered so far for the estimation of per flow volume.

Difficulties for the modeling arise from the fact that sampling is done before classification and the number of packets per flow in the population is unknown. Sampling here is done before classification, because classification of every single packet is problematic at high speed interfaces. Furthermore, input parameters for the accuracy calculation need to be derived from the stored information in the Cisco flow cache. Due to memory limitations only aggregated information is available.

4.4 Case Differentiation

In 1.2 three dimensions for the assessment of selection schemes were introduced: Costs, accuracy and available traffic information. In the following the dependencies between costs, represented by fraction of data that has to be processed, and accuracy, represented by bias and precision of the estimate, are investigated by mathematical modeling. The dimension of traffic information is addressed by differentiating cases with different amount of available information.

In accordance to 4.3.2 only count-based schemes are considered. That means one basic assumption is that link counters, which count the number of all incoming packets on an interface, are available. For the estimation of the flow volume a classification of packets into flows is needed. The number of packets per flow is a very useful additional piece of information for the estimation of the flow volume. But the number of packets per flow can only be examined after classification of the whole population into flows. Monitoring the number of packets per flow is only possible if classification is done before packet selection and additional counters (one per flow) are provided and updated. Nevertheless, as explained in section 2.3.2.2 it is of advantage to perform sampling before classification. In such cases only the selected packets have to be classified, but the exact packet count per flow cannot be determined. Therefore, the following cases have to be distinguished:

- Case A: Number N_f of packets per flow is known
- Case B: Number N_f of packets per flow is unknown

Since the available traffic information is a relevant parameter for the calculation of the estimation accuracy, both cases are considered separately.

4.5 Volume Estimation with n-out-of-N Sampling, Case A

4.5.1 Initial Assumptions

The following initial assumptions are made in order to simplify the mathematical modeling:

Assumption 1: *Link counter, which count all packets received on an interface, are available and the measurement interval length is given in number of packets.*

Assumption 2*: Flow timeouts are not considered. A flow is considered as the sequence of packets that have common properties (defined by the flow keys) independent of the time between packet arrivals (inter-packet time) and of the observation of packets that belong to other flows. So each packet observed during the measurement interval that has the flow properties is considered as part of this single flow.*

4.5.2 Mathematical Model

4.5.2.1 Special Case: N_f=N

Let us first consider the simplest case that all packets in the measurement interval belong to the same flow, i.e., N_f=N. If one selects n packets out of the N packets in the measurement interval, one can be sure that all selected packets belong to the same flow, i.e., the number of packets n_f in the sample that belong to flow f equals the number of selected packets (n_f=n). Since the sampling parameters N and n are configured in advance, one gets a fixed sample fraction[15], i.e., for each sample run with the given parameters the same sample fraction is applied.

$$f_{R,f} = \frac{n_f}{N_f} = \frac{n}{N} = f_R \tag{4.3}$$

The sample fraction for flow f is equal to the configured overall sample fraction. The volume (sum of observed bytes) from flow f can be estimated as follows:

$$\hat{Sum}_f = \hat{Sum} = N \cdot \hat{\mu}_x = \frac{N}{n} \cdot \sum_{i=1}^{n} x_i \tag{4.4}$$

[15] The symbol f_R here is used to indicate the real sample fraction. Later it will be distinguished between the real sample fraction f_R and the target sample fraction f_T.

Here $\hat{\mu}_x$ denotes the estimated mean packet size and x_i denotes the number of bytes in the i^{th} packet of the sample. This estimation method, the method of moments, provides an unbiased estimate for the mean estimation. Therefore one gets an unbiased estimate for the sum:

$$E\left[\hat{Sum}\right] = E\left[N \cdot \hat{\mu}_x\right] = N \cdot E\left[\hat{\mu}_x\right] = N \cdot \mu_x = Sum \tag{4.5}$$

The standard error for the random selection of n packets out of N packets can be calculated as follows (see e.g., [Schw91]):

$$StdErr_{abs}\left[\hat{Sum}\right] = N \cdot \frac{\sigma_x}{\sqrt{n}} \cdot \sqrt{\frac{N-n}{N-1}} \tag{4.6}$$

Here σ_x denotes the standard deviation of the packet sizes in the measurement interval. The standard error and with this the estimation accuracy depends on the following parameters:

- sample size *n*
- population size *N*
- standard deviation of the packet sizes σ_x

The sampling parameter *N* and *n* are configured in advance. σ_x is a traffic characteristic and usually unknown. Therefore one need to estimate it from the values in the sample, assume a maximum value (e.g., derived from the maximum transmission unit MTU) or use knowledge from previous measurements to predict this value. If the sample size is small compared to the population size (*n/N<5%*), one can neglect the finite population correction factor (last factor) and get:

$$StdErr_{abs}\left[\hat{Sum}\right] = N \cdot \frac{\sigma_x}{\sqrt{n}} \tag{4.7}$$

4.5.2.2 General Case: N_f <N

If only some packets in the measurement interval belong to the flow of interest, the number of packets N_f in flow *f* is smaller than the parent population (N_f<*N*). If N_f is known, it can be considered as the parent population and the number n_f of packets that were selected from this flow as sample size. That means one can apply sampling per flow and simply models the n-out-of-N sampling as n_f-out-of N_f sampling. Then the same formulas can be applied as seen above, simply by substituting *N* by N_f and *n* by n_f.

$$\hat{Sum}_f = N_f \cdot \hat{\mu}_{x_f} = \frac{N_f}{n_f} \cdot \sum_{i=1}^{n_f} x_{i,f} \tag{4.8}$$

Here N_f denotes the number of packets from flow *f* in the parent population, $\hat{\mu}_{x_f}$ denotes the estimated mean packet size for flow *f*, n_f denotes the number of packets from flow *f* in the sample and $x_{f,i}$ denotes the number of bytes in the i^{th} packet from flow *f* in the sample. One gets an unbiased estimation:

$$E\left[\hat{Sum}_f\right] = E\left[N_f \cdot \hat{\mu}_{x_f}\right] = N_f \cdot E\left[\hat{\mu}_{x_f}\right] = N_f \cdot \mu_{x_f} = Sum_f \tag{4.9}$$

The standard error can be calculated with:

$$StdErr_{abs}\left[\,\hat{Sum}_f\,\right] = N_f \cdot \frac{\sigma_{x_f}}{\sqrt{n_f}} \cdot \sqrt{\frac{N_f - n_f}{N_f - 1}} \tag{4.10}$$

The standard error depends on the following parameters:

- the number n_f of packets from flow *f* in the sample
- the number N_f of packets from flow *f* in the parent population
- the standard deviation σ_{x_f} of the packet sizes in the flow

4.6 Volume Estimation with n-out-of-N Sampling, Case B

4.6.1 Initial Assumptions

In addition to the initial assumption stated in section 4.5.1 the following assumptions are made, in order to be able to derive a model for case B.

Assumption 3*: The sample size n is small compared to the parent population N and fulfils the condition*

$$\frac{n}{N} \leq 0.05 \tag{4.11}$$

Assumption 4*: The proportion P_f of packets N_f of the investigated flow to all packets N in the traffic mix lies between 0.1 and 0.9*

$$0.1 < P_f < 0.9 \quad \text{with} \quad P_f = \frac{N_f}{N} \tag{4.12}$$

Assumption 5*: The number of packets in the sample is larger than 30*

$$n > 30 \tag{4.13}$$

With those assumptions one can approximate the hyper geometrical distribution *Hy(N, N_f, n)* by a binomial distribution *B(n, N_f/N)* (see e.g., [Schw91]). Furthermore a large sample size allows to assume a normal distribution for the estimated mean. Those assumptions reduce the complexity of the problem space and are necessary to allow the derivation of a mathematical model. Nevertheless, it has to be investigated whether and to what degree the assumptions hold true in the specific cases in reality. Furthermore, it is investigated with experiments on real traces if and how empirical results depart from the model for cases where the assumptions do not hold.

4.6.2 Mathematical Model

4.6.2.1 Special Case: N_f=N

For case B the number N_f of packets per flow is unknown. Nevertheless, the number *N* of all packets in the measurement interval is a preconfigured sampling parameter and therefore

known. That means in the special case where all packets belong to only one flow, i.e., N_f=N, the same model as for case A (4.5.2.1) can be applied.

4.6.2.2 General Case: N_f<N

In the general case N_f is smaller than the population size N. If a sample of n packets is selected out of the population, it can contain packets from different flows. The number n_f of packets in the sample that belong to flow f can vary for each sample run and is unknown before the selection process. For case A the sample fraction could be derived after completion of the selection process with n_f and N_f. An estimate for the flow volume could be calculated as shown in (4.8).

For case B, N_f is unknown. That means even after the sampling process has completed, the sampling fraction $f_{R,f}$ per flow cannot be calculated. The fact that the number of packets of the flow is not known, introduces a lot more complexity into the problem space. It moves the task from a simple mean estimation for one random variable to the estimation of parameters of a compound distribution with two different random variables. To approach a solution for the problem, N_f can be estimated by using the proportion of packets that belong to flow f in the sample as follows:

$$\hat{N}_f = \frac{n_f}{n} \cdot N \tag{4.14}$$

The flow volume of flow f can be estimated as follows:

$$\hat{Sum}_f = N_f \cdot \hat{\mu}_{x_f} = \frac{N}{n} \cdot \sum_{i=1}^{n_f} x_{i,f} \tag{4.15}$$

That means an estimate for the flow volume can be calculated. Nevertheless, in contrast to case A, here the standard formulas for expectation and standard error as used in 4.5.2.2 can not be applied. The reason for this is that the estimate now contains two random variables, n_f and $x_{i,f}$. In order to assess the estimation quality one has to look at the expectation and variance of a sum of random variables, where the number of addends itself is a random variable.

$$E\left[\hat{Sum}_f\right] = E\left[\frac{N}{n} \cdot \sum_{i=1}^{n_f} x_{i,f}\right] = \frac{N}{n} \cdot E\left[\sum_{i=1}^{n_f} x_{i,f}\right] \tag{4.16}$$

$$V\left[\hat{Sum}_f\right] = V\left[\frac{N}{n} \cdot \sum_{i=1}^{n_f} x_{i,f}\right] = \frac{N^2}{n^2} \cdot V\left[\sum_{i=1}^{n_f} x_{i,f}\right] \tag{4.17}$$

The problem space can be further narrowed by making use of knowledge about the random variables. $x_{i,f}$ denotes the number of bytes of the i^{th} selected packet. Since a random selection is applied, it can be assume that the $x_{i,f}$ are statistically independent random variables (RVs). The packet size is a discrete value. Therefore the $x_{i,f}$ are discrete RVs. Furthermore, all RVs $x_{i,f}$ of flow f have the same distribution, the distribution of the packet sizes for flow f. So the $x_{i,f}$ of one flow can be considered as independent identical distributed (i.i.d) RVs. The shape

of the distribution is unknown, but one can denote the expectation and variance with the moments of the packet size distribution for flow *f*.

$$E\left[x_{1,f}\right] = E\left[x_{2,f}\right] = \ldots = E\left[x_{n_f,f}\right] = \mu_{x_f} \tag{4.18}$$

$$V\left[x_{1,f}\right] = V\left[x_{2,f}\right] = \ldots = V\left[x_{n_f,f}\right] = \sigma^2_{x_f} \tag{4.19}$$

n_f is also a discrete random variable and can take the values n_f*=0, 1,2,…, n*. n_f can be modeled as number of hits ("packet belongs to flow f") in *n* trials (selection of a packet from the population). The packet sampling is a selection without replacement, because a packet that once has been selected cannot be selected again. Therefore n_f follows a hyper geometric distribution. But with the assumptions stated in 4.6.1 the hyper geometric distribution can be approximated by a binomial distribution. The probability that the selected packet belongs to flow f (i.e., probability of success) is equal to the proportion N_f/N of packets from flow *f* in the population.

$$n_f \sim B\left(n, \frac{N_f}{N}\right) \tag{4.20}$$

The expectation and variance of n_f is given by the standard formulas for a binomial distribution:

$$E\left[n_f\right] = n \cdot \frac{N_f}{N} \tag{4.21}$$

$$V\left[n_f\right] = n \cdot \frac{N_f}{N} \cdot \left(1 - \frac{N_f}{N}\right) \tag{4.22}$$

With the considerations above, the task is reduced to the calculation the expectation and variance of a random variable *Z*, where *Z* is the sum of independent identical distributed (i.i.d.) random variables *X* for which the number of summands *Y* is a binomial distributed random variable. A formula to calculate the expectation of such a random variable can be found in [Fisz63].

$$E\left[Z\right] = E\left[X\right] \cdot E\left[Y\right] \qquad \text{for} \quad Z = \sum_{i=1}^{Y} X_i \tag{4.23}$$

With this the expectation of the estimated volume is calculated as follows:

$$E\left[\hat{Sum}_f\right] = \frac{N}{n} \cdot E\left[\sum_{i=1}^{n_f} x_{i,f}\right] = \frac{N}{n} \cdot E\left[x_{i,f}\right] \cdot E\left[n_f\right]$$

$$= \frac{N}{n} \cdot \mu_{x_f} \cdot n \cdot \frac{N_f}{N} = N_f \cdot \mu_{x_f} = Sum_f \tag{4.24}$$

The expectation of the estimate equals the real volume. That means the estimation is unbiased.

A formula to calculate the variance for this special case, but for continuous RVs is derived in [WeOw75]. The derivation for discrete variables can be found in the appendix section E.

$$V[Z] = E[Y] \cdot V[X] + E[X]^2 \cdot V[Y] \quad \text{for} \quad Z = \sum_{i=1}^{Y} X_i \tag{4.25}$$

With the above formula the variance of the estimated flow volume can be expressed as follows:

$$V\left[\hat{Sum}_f\right] = \frac{N^2}{n^2} \cdot V\left[\sum_{i=1}^{n_f} x_{i,f}\right] = \frac{N^2}{n^2} \cdot \left(E[n_f] \cdot V[x_{i,f}] + E[x_{i,f}]^2 \cdot V[n_f]\right) \tag{4.26}$$

With (4.18), (4.19), (4.21) and (4.22) one gets:

$$\begin{aligned} V\left[\hat{Sum}_f\right] &= \frac{N^2}{n^2} \cdot \left(n \cdot \frac{N_f}{N} \cdot \sigma_{x_f}^2 + \mu_{x_f}^2 \cdot n \cdot \frac{N_f}{N} \cdot \left(1 - \frac{N_f}{N}\right)\right) \\ &= \frac{N^2}{n} \cdot \left(\frac{N_f}{N} \cdot \sigma_{x_f}^2 + \mu_{x_f}^2 \cdot \frac{N_f}{N} \cdot \left(1 - \frac{N_f}{N}\right)\right) \end{aligned} \tag{4.27}$$

It can be seen that the variance of the estimated flow volume, and with this the expected accuracy of the estimation depends on the following parameters:

- the sample size *n*
- the population size *N*
- the number N_f of packets from flow *f* in the population
- the mean μ_{x_f} of the packet sizes in the flow
- the variance $\sigma_{x_f}^2$ and of the packet sizes in the flow

The sample size n and the population size N are preconfigured sampling parameters. N_f, μ_{x_f} and $\sigma_{x_f}^2$ are aggregated flow characteristics. μ_{x_f} and $\sigma_{x_f}^2$ are parameters of the packet size distribution in the flow.

The absolute and the relative standard error can be derived as follows.

$$StdErr_{abs}\left[\hat{Sum}_f\right] = \sqrt{V\left[\hat{Sum}_f\right]} = \sqrt{\frac{N^2}{n} \cdot \left(\frac{N_f}{N} \cdot \sigma_{x_f}^2 + \mu_{x_f}^2 \cdot \frac{N_f}{N} \cdot \left(1 - \frac{N_f}{N}\right)\right)} \tag{4.28}$$

$$StdErr_{rel}\left[\hat{Sum}_f\right] = \frac{StdErr_{abs}\left[\hat{Sum}_f\right]}{Sum_f} = \frac{\sqrt{\frac{N \cdot N_f}{n} \cdot \left(\sigma_{x_f}^2 + \mu_{x_f}^2\right) - \frac{N_f^2}{N} \cdot \mu_{x_f}^2}}{N_f \cdot \mu_{x_f}} \tag{4.29}$$

Furthermore, as noted in 4.6.2.1 one can see that one gets the same model as for Case A in 4.5.2.1, if all packets in the measurement interval belonged to one flow (i.e., N_f=N).

$$StdErr_{abs}\left[\hat{Sum}_f\right] = \sqrt{\frac{N^2}{n} \cdot \left(\frac{N}{N} \cdot \sigma_{x_f}^2 + \mu_{x_f}^2 \cdot \frac{N}{N} \cdot \left(1 - \frac{N}{N}\right)\right)} = \sqrt{\frac{N^2 \cdot \sigma_{x_f}^2}{n}} = N \cdot \frac{\sigma_{x_f}}{\sqrt{n}} \tag{4.30}$$

Since the assumption was made that *n/N<5%* in order to derive formula (4.28), one here gets formula (4.7) with the neglected finite population correction factor as a result.

4.6.3 Parameter Dependencies for n-out-of-N, Case B

Since formula (4.28) is comparatively complex, it is shown here how the estimation accuracy depends on the different parameters.

4.6.3.1 Dependency on the Sampling Fraction

It is easy to see from the formula that the variance decreases if the sample size increases. It is quite obvious that a higher sample size leads to a higher accuracy. For a sample size of 0 one would get an infinite variance, which means an accuracy of 0. If the whole parent population is sampled (n=N), one would expect a variance or standard error of 0. In order to investigate how the standard error dependents on the sampling fraction, the following parameter are set as fixed and the sampling fraction, which is calculated as ratio of sample size and population size, is varied:

$$f_R = \frac{n}{N} \tag{4.31}$$

Fixed Parameters	**Value**
Measurement Interval length (N)	100,000
Packets from flow f (N_f)	12,000
Mean packet size within flow f	250 Bytes
Standard deviation of packet sizes within flow f	50 Bytes
Calculated from parameters	
Proportion of packets from flow f (P_f)	0.12 (12 %)
Real total volume of packet from flow f in parent population	3,000,000 Bytes

Table 4-3: Parameter Settings

Figure 4-1 shows the dependency of the relative standard error of the estimate from the sampling fraction. The first diagram shows the expected values for the whole range up to a sample fraction of 100%. The red marked field marks the range for which assumption 3 does not hold (n/N>5%). The second diagram shows more detailed the expected values for the range where assumption 3 holds (n/N< 5%).

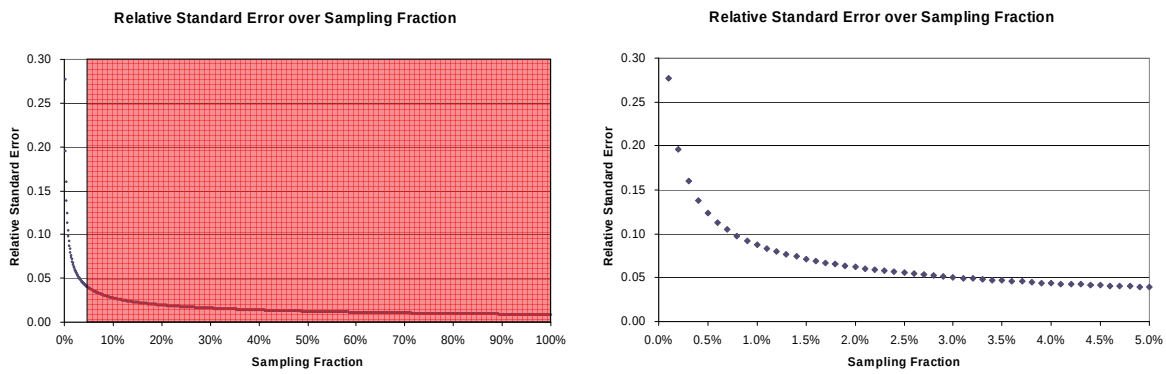


Figure 4-1: Dependency of the Relative Standard Error from the Sampling Fraction

As expected the standard error decreases for larger sampling fractions. It also can be seen that there is quite a big range of sample fractions for which one cannot rely on the model, because initial assumptions are violated. It is later checked with experiments on real network traffic if and how empirical results differ from the model in that range. At least it is expected that the standard error is reduced to zero if 100% (i.e., all packets) are sampled.

The curve for the absolute standard error can be derived from this by simply multiplying each value in the curve with the constant real volume of the flow. Therefore the shape of the curve would look the same.

4.6.3.2 Dependency on the Measurement Interval Length

All packets that are observed in the measurement interval form the population size N. For count-based n-out-of-N sampling the measurement interval length is measured in number of packets. Therefore here *N* also describes the measurement interval length. If the measurement interval length is increased, one has to take into account that other parameters are also changed because they depend on the population size *N*. The parameters that can depend on *N* are the following

- Sampling fraction *n/N*
- Packet proportion P_f
- Mean packet size μ_{x_f}
- Standard deviation σ_{x_f} of the packet size

With regard to the sampling fraction one has to distinguish whether:

a) *n/N=const*: The sampling fraction remains constant. That means that the number of sampled packets is increased if *N* is increased or

b) *N=const.*: The sample size *n* remains constant. That means the sampling fraction automatically decreases if *N* becomes larger.

With regard to the proportion P_f of packets from flow *f* one has to distinguish whether:

c) P_f=*const*: The proportion of packets from flow *f* remains constant. Therefore the number of packets from flow *f* increases with the population *N*. (P_f=*const.* ➔ N_f increases also)

d) N_f=*const*: The number of packets from flow *f* remains constant. Therefore the proportion P_f has to decreases if *N* increases. In this case only packets that do not belong to flow *f* are added to the measurement interval.

For the following graphical presentation it is assumed that the sampling fraction remains constant (case a). That means if the measurement interval is enlarged, also the sample size *n* is increased to keep the fixed sampling fraction. Furthermore, it is assumed that the proportion of packets from flow *f* remain constant if the measurement interval is enlarged (case c). That means that number of packets N_f from flow *f* increases for larger MIs. It is also assumed that mean and standard deviation of the packet sizes within the flow remain the same. The following parameters are used for the graphical presentation:

Fixed Parameters	**Value**
Sampling fraction	5%
Proportion of packets from flow f (P_f)	0.12 (12 %)
Mean packet size within flow f	250 Bytes
Standard deviation of packet sizes within flow f	50 Bytes

Table 4-4: Parameter Settings

Figure 4-2 shows how the absolute and relative standard error evolves if the measurement interval length *N* increases.

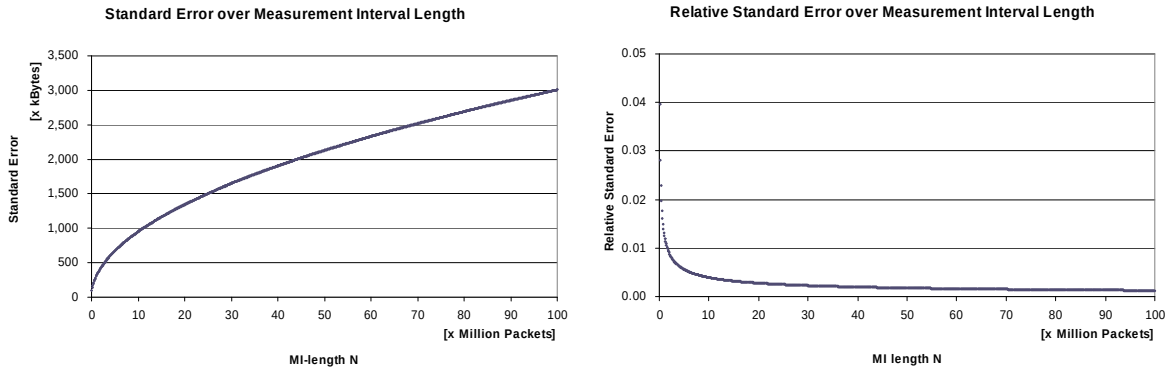


Figure 4-2: Dependency of the Standard Error from the Measurement Interval Length

One would expect that the standard error decreases (accuracy increases) if a longer measurement interval is chosen and the sampling fraction and the proportion of packets from flow f remain constant. Nevertheless, as can be seen in Figure 4-2 the absolute error increases. This is due to the fact that with a larger measurement interval (with fixed proportion P_f) also more packets from flow *f* are observed and therefore also the flow volume for flow *f*

increases. If one looks at the relative standard error (normalized by the real flow volume) one can see the expected decrease of the error for larger measurement intervals.

The result becomes clearer if one considers the following: If the sampling fraction remains constant, the number n of selected packets has to increase if the population size N increases. If also the proportion of packets from flow f remains constant, the number N_f of packets from flow f also has to increase if N increases. Therefore it is likely that also the number n_f of packets from flow f in the sample increases. With a higher number of packets from flow f in the sample, a higher estimation accuracy can be expected.

4.6.3.3 Dependency on the Packet Proportion from Flow f

The proportion P_f of packets from flow f is the fraction of packets from flow f in the population and can take values between 0 and 1.

$$P_f = \frac{N_f}{N} \qquad 0 \le P_f \le 1 \tag{4.32}$$

In the formula for the variance of the estimated flow volume contains two terms with P_f.

$$V\left[\hat{Sum}_f\right] = \frac{N^2}{n} \cdot \left(\left(\sigma_{x_f}^2 + \mu_{x_f}^2 \right) \cdot P_f - \mu_{x_f}^2 \cdot P_f^2 \right) \tag{4.33}$$

The first term $\left(\sigma_{x_f}^2 + \mu_{x_f}^2 \right) \cdot P_f$ expresses a linear relation between the variance of the estimate and the proportion P_f. The second term $\mu_{x_f}^2 \cdot P_f^2$ describes a quadratic dependency. The weights of each of these components depend on the variance and the mean of the packet sizes in the flow.

For the case that all packets belong to flow f ($P_f=1$, $N_f=N$) one gets the following variance:

$$V\left[\hat{Sum}_f\right] = N^2 \cdot \frac{\sigma_{x_f}^2}{n} \tag{4.34}$$

As one would expect, this equals exactly the variance one gets for simple random sampling of n elements from a population with N elements. For the case that no packet belongs to flow f ($P_f=0$, $N_f=0$) the variance becomes 0.

$$V\left[\hat{Sum}_f\right] = 0 \tag{4.35}$$

If an estimate has no variation describes the maximum accuracy one can get, because in each sample run one would get the same estimate as a result. It is obvious that the variance gets 0 for $P_f=0$. If the proportion is 0, one can never observe any packet from flow f in the sample. The number n_f of packets of flow f in the sample will always be 0, and with this the estimated sum always equals 0, which is a perfect estimation of the real sum.

In order to show how the accuracy depends on the proportion of packets from flow f in the trace the following parameters are used:

Fixed Parameters	Value
Sampling fraction (n/N)	5%
Measurement Interval length (N)	100,000
Mean packet size within flow f	250 Bytes
Standard deviation of packet sizes within flow f	50 Bytes

Table 4-5: Parameter Settings

Figure 4-3 shows the dependency of the absolute and relative standard error from the proportion of packets from flow *f*. In the red shaded areas at least one of the assumptions does not hold. The values in these sections should be treated with care, because the formula may not be applicable in these cases. One can check with empirical investigations what accuracy can be achieved if the proportion falls into these areas.

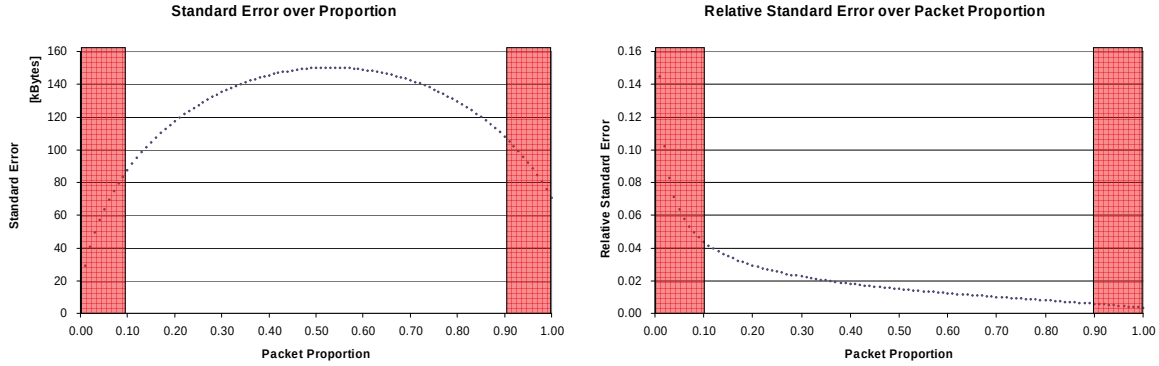


Figure 4-3: Dependency of the Standard Error from the Proportion of Packets

As expected the absolute standard error approaches 0 if the proportion gets smaller. Furthermore, if all packets of the population belong to flow *f* ($P_f=1$) one gets the standard error that one would expect for an n-out-of-N random sampling without distinguishing flows.

The absolute standard error has its maximum if the proportion of packets from flow *f* is 50% of the overall traffic. An explanation for this is that the variance of the binomial distributed random variable n_f has its maximum if the success probability is 0.5 (which is the case for a packet proportion of 50%).

This effect can be explained by following example. If 10 balls are selected from a population of 50 black and 50 white balls in order to estimate the number of black balls, the results for each sample run can differ very much. This is the case, because it is possible to get any number from 0 to 10 black balls in the sample, so the estimate can vary between 0 and 10. If only 1 ball in the population was black, the estimate could not vary very much because the number of black balls in the sample is either be 0 or 1. So the number of packets that one gets from flow *f* in the sample would vary most if the flow has a proportion of 0.5 of the traffic. Since the variance of n_f is part of the formula, this leads to a higher variance (and standard error) for the volume estimation.

The relative standard error is calculated by dividing the absolute standard error by the real flow volume. The real volume increases if N_f increases. The second diagram shows how the relative standard error depends on the packet proportion. The volume increases linear with the packet proportion. So the curve for the relative error is the result of the division of the curve for the absolute error by a linear function.

4.6.3.4 Dependency on the Packet Size Mean in Flow f

The variance of the volume estimates for a flow volume depends on the mean packet size within the flow. For the diagram below the following values are set:

Fixed Parameters	**Value**
Sampling fraction (n/N)	5%
Measurement Interval length (N)	100,000
Proportion of packets from flow f (P_f)	0.12 (12 %)
Standard deviation of packet sizes within flow f	50 Bytes

Table 4-6: Parameter Settings

Figure 4-4 shows how the standard error of the estimate depends on the mean packet size of the flow.

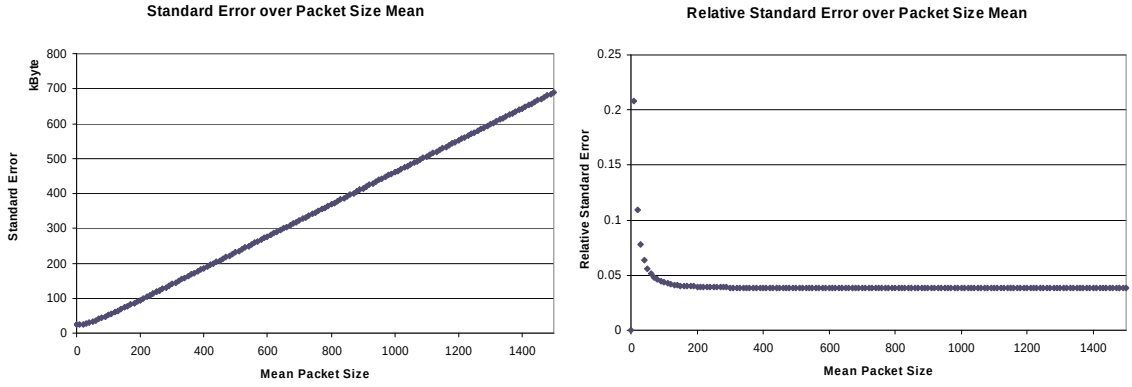


Figure 4-4: Dependence of Standard Error from Mean Packet Size in Flow

The absolute standard error increase for larger mean values, because the overall volume increases and with this a larger absolute error can be made. For a mean packet size of 0 the diagram shows a theoretical standard error of 24 kbytes. But this point cannot be reached in reality. For the diagrams the packet size variance is assumed to be constant. But both, mean and standard deviation of the packet sizes, depend on the individual packet sizes in the flow. If the mean packet size is 0, all packets have to have size 0 and therefore the standard deviation also would be 0 (and not 50 bytes as assumed). That means the case that the standard deviation is larger than 0 if the packet size mean is 0 cannot occur in reality.

The second diagram shows how the relative standard error depends on the mean packet size. Since packet proportion and MI length are assumed to remain constant, the flow volume

increases linear with the mean packet size ($Sum_f = N_f \cdot \mu_f$ with $N_f = P_f \cdot N$). Therefore the curve for the relative error is the result of the division of the curve for the absolute error by a linear function.

4.6.3.5 Dependency on the Standard Deviation of Packet Sizes in Flow f

The variance of the volume estimates for a flow volume also depends on the standard deviation of the packet sizes within the flow. For the diagram below the following values are fixed:

Fixed Parameters	**Value**
Sampling fraction (n/N)	5%
Measurement Interval length (N)	100,000
Proportion of packets from flow f (P_f)	0.12 (12 %)
Mean of packet sizes within flow f	250 Bytes

Table 4-7: Parameter Settings

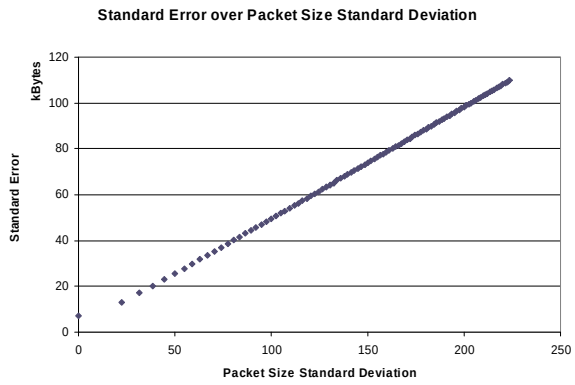


Figure 4-5: Dependence of (Absolute) Standard Error from Packet Size Standard Deviation in Flow

The higher the standard deviation of the packet sizes within the flow the more different estimates are possible. Therefore the expected standard error increases if the standard deviation of packets in the flow increases. Since the mean is set constant for this considerations, the real flow volume also would be constant. Therefore the shape of the curve for the relative error would look the same, because it is the result of a division by a constant.

4.6.3.6 Conclusion

From the theoretical investigations it can be seen that the estimation accuracy depends on sampling parameters and flow characteristics. A better accuracy (i.e., a lower relative standard error) can be achieved if:

- the sample fraction (*n/N*) is high

- the measurement interval length N is longer and the traffic characteristics remain more or less constant (see conditions in 4.6.3.2)

Furthermore one gets a higher accuracy for flows with the following characteristics:

- many packets (i.e., packets of the flow are a large proportion P_f of the population)
- large mean packet size μ_{x_f} (i.e., rather big packets in the flow)
- small packet size variance σ_{x_f} (i.e., packet sizes differ not so much)

4.7 Volume Estimation with Systematic Count-based Sampling

4.7.1 Initial Assumptions

All assumption stated in section 4.5.1 for case A and additional the assumptions of section 4.6.1 for case B are assumed to hold true. Furthermore, it is assumed that the distance between subsequent samples remain equal.

Assumption 7*: The sampling period K (number of packets) is constant in the measurement interval.*

$$K = const. \tag{4.36}$$

Assumption 8*: The number of packets N in the measurement interval is a multiple of K*

$$N = L \cdot K \tag{4.37}$$

4.7.2 Mathematical Model

In systematic count-based sampling every K^{th} packet is selected. The estimate is calculated in the same way as for n-out-of-N sampling by extrapolating the sum of the packet sizes in the sample with the sample fraction (see section 4.5 for case A and section 4.6 for case B)..

If all packet sizes and flow IDs in the measurement interval are independent, the systematic selection does not differ from a random selection. In such a case the same mathematical model as for n-out-of-N sampling can be applied and one gets the same estimation accuracy as for n-out-of-N sampling as described in section 4.5 (if N_f is known) or section 4.6 (if N_f is unknown).

Nevertheless, if there are correlations between packet sizes in the measurement interval, the systematic selection process can interfere with periodicities in the packet sequence. In such cases one may get a non-representative accumulation of packets with specific properties (e.g., too many large packets or too many packets from a specific flow) in the sample. This leads to a biased estimation. The nature of this bias heavily depends on the position of packets from flow f in the packet sequence. That means in addition to the parameters N_f, μ_{x_f} and σ_{x_f} mentioned above one here needs to consider the packet arrival sequence.

Flows with the same number of packets and packet size distributions can be distributed in different ways over the measurement interval. Therefore one cannot derive a generic model, which is valid for arbitrary traces, as for the random selection methods.

If not all packets in the measurement interval belong to one flow ($N_f<N$), one needs to consider potential correlations and periodicities of two packet attributes:

- Correlations and periodicities in the occurrence of packet sizes in the measurement interval
- Correlations and periodicities in the occurrence of flow IDs (membership of packets to flows, derived from packet content) in the measurement interval.

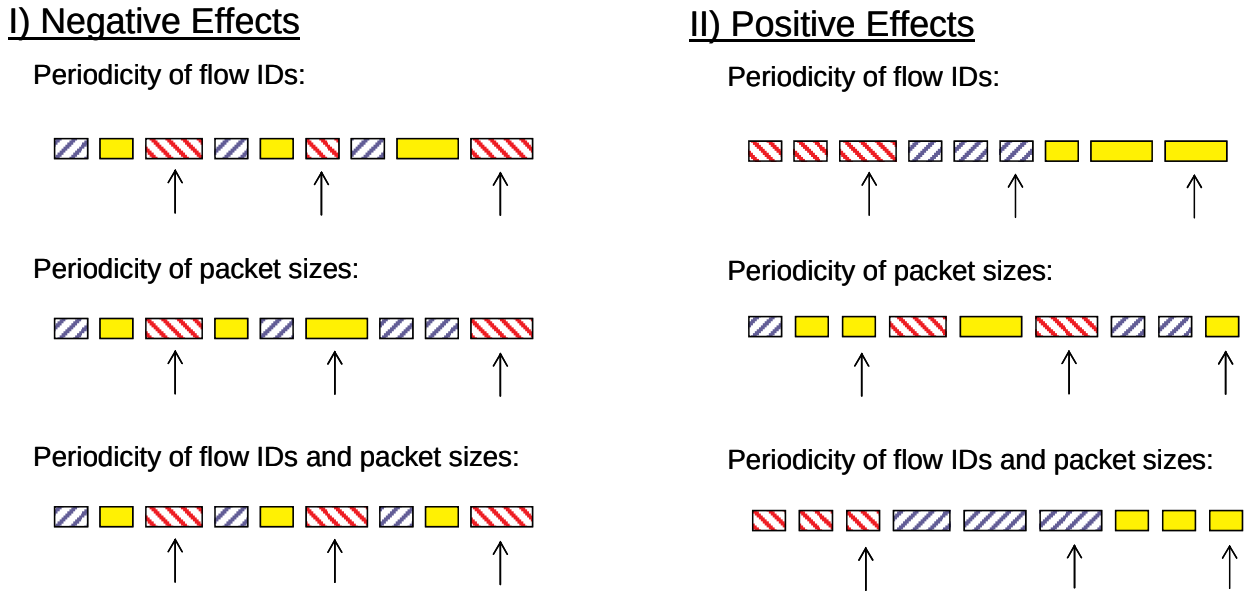


Figure 4-6: Systematic Traffic Structures

Figure 4-6 illustrates the different cases. The color of the rectangles indicate the flow ID (the flow to which the packet belongs) and the size of the rectangles indicate the packet sizes. One easily can see that systematic structures in the traffic can lead to a biased selection with systematic sampling. In some cases (I) one gets negative effects due to a non-representative selection of packets in the sample. In other cases (II) one gets a better accuracy than for random sampling, because the selected packets are more distributed over the whole measurement interval. This is the case if packets with equal attributes arrive in bursts at the observation point (Figure 4-6, case II). In the special case that $N_f=N$, all flow IDs are equal and one only needs to consider systematic structures of the packet sizes.

Knowing the number N_f of packets that belong to flow f (case A) may help to assess (after the sampling) whether a selection was biased towards flow IDs or not. E.g., if one knows that the majority of packets belong to one flow and the percentage of packets from that flow in the sample is small, the selection is biased with regard to the flow IDs. In case B the number N_f of

packets from the flow of interest is not known. Therefore one does not even know what percentage of packets from flow *f* was selected.

4.8 Volume Estimation with Count-based Stratified Sampling (1-in-K Sampling)

1-in-K sampling[16] is a count-based stratified n-out-of-N sampling. In 1-in-K sampling one also selects *n* out of *N* packets. But the selection process is done in two steps. First the measurement interval is grouped into subintervals of size K and then the random selection is done per subinterval.

The measurement interval, i.e., the population for which a parameter should be estimated, still consists of *N* packets. Therefore 1-in-K sampling cannot simply be interpreted as n-out-of-N sampling with *n=1* and a measurement interval length of *K* packets. Because the estimate is not calculated from the one packet selected in the sub interval, but from all packets that were selected in all subintervals in the measurement interval.

The 1-in-K technique is used to distribute samples over the measurement interval. The measurement interval is split into *L* smaller subintervals. Then a 1-in-K selection is performed for each subinterval. An estimate is calculate for the whole measurement interval from the L samples from the *L* intervals (in our case $L=n$).

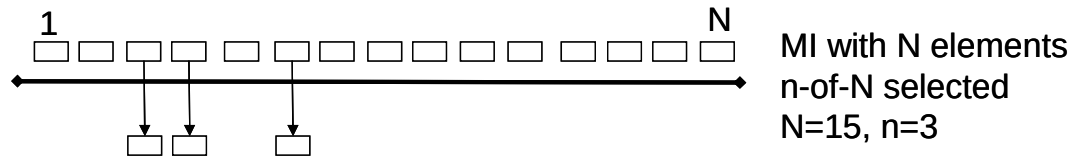
In order to distinguish between measurement interval and subintervals the following notation is used:

- MI denotes the measurement interval for which an estimate is calculated
- *N* denotes the number of elements in the population, which is used as basis for the estimation.
- *n* denotes the number of samples selected from the population N
- *K* denotes the number of elements in a subinterval.
- *k* denotes the number of samples from a subinterval (here *k* is always 1)
- *L* denotes the number of subintervals within the MI. For 1-in-K sampling, the number of samples *n* is equal to the number of subintervals. $L=N/K$

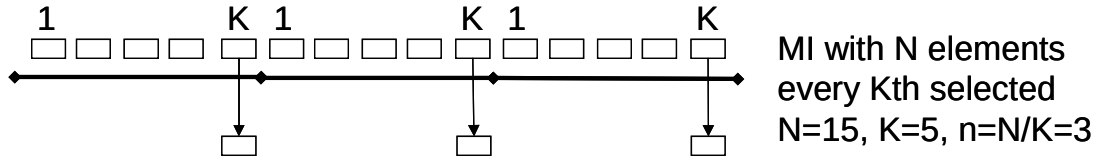
1-in-K sampling distributes the selected elements over the trace. But in contrast to systematic sampling, which distributes the samples equally over the trace, some random element is preserved by doing a random selection within the subintervals. The 1-in-K sampling method is a count-based stratified sampling where the subintervals form the different strata and only 1 element is selected per stratum. Figure 4-7 compares the different methods.

[16] A commonly used name for his method is 1-in-N sampling. In order to avoid confusion with the measurement interval length it was decided to call the scheme 1-in-K sampling.

Random n-out-of-N:



Systematic:



Random 1-in-K:

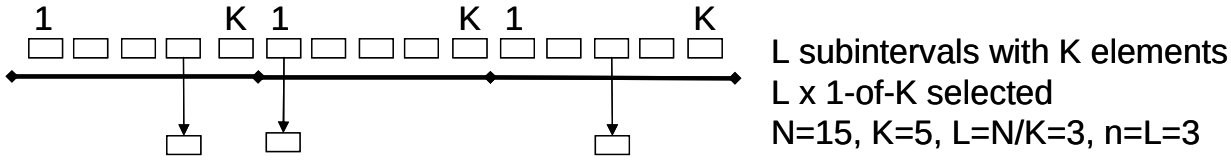


Figure 4-7: Comparison of Sampling Schemes

The population is stratified into L strata. All strata have an equal size (K). Only 1 element is selected per stratum. Since for each stratum the same number of elements (only 1 packet) is selected, the selection is done in accordance to the equal allocation scheme. Since the sample fraction ($k/K=1/K$) is the same for each subinterval and equals n/N (since $n = L \cdot k$ and $N = L \cdot K$) it is also a proportional allocation (see 3.4.4).

4.8.1 Mathematical Model

Finding a model for 1-in-k Sampling is approached in two steps. First the special case where all packets belong to the same flow ($N_f=N$) is considered. Then the case for multiple flows ($N_f<N$) is considered.

4.8.1.1 Initial Assumptions

All assumption as stated in section 4.5.1 for case A and the assumptions of section 4.6.1 for case B are assumed to hold true. Furthermore, the following additional assumptions are made.

Assumption 7*: The number K of packets in a subinterval is constant and remains the same for all subintervals in the measurement interval.*

$$K = const. \tag{4.38}$$

Assumption 8*: The number of packets N in the measurement interval is a multiple of K*

$$N = L \cdot K \tag{4.39}$$

4.8.1.2 Special Case (N_f=N)

For the first case the whole traffic mix (i.e., all packets in the measurement interval) is considered as one single flow (N_f=N). In the single flow case, one gets the following absolute standard error for random n-out-of-N sampling (see section 4.5)

$$StdErr[\hat{Sum}]_{rand} = N \cdot \frac{\sigma_x}{\sqrt{n}} \tag{4.40}$$

The standard error for n-out-of-N sampling depends on the number N of packets in the measurement interval, the sample size n and the variance of packet sizes within the measurement interval.

If stratified sampling is used the standard error is calculated as follows [see Coch72, following equation 5.9]:

$$StdErr[\hat{Sum}]_{strat} = \sqrt{\sum_{l=1}^{L} N_l \cdot \left(N_l - n_l\right) \cdot \frac{\sigma_l^2}{n_l}} \tag{4.41}$$

where L is the number of strata, N_l is the number of elements in stratum l, n_l is the number of selected elements from stratum l and σ_l^2 is the variance of the survey variable (packet sizes) in the l^{th} stratum. In our case all strata have the same amount of elements K:

$$N_1 = N_2 = \ldots = N_L = \ldots = N_L = K \tag{4.42}$$

Furthermore from each stratum the same amount $k=1$ of packets is selected:

$$n_1 = n_2 = \ldots = n_L = \ldots = n_L = k = 1 \tag{4.43}$$

Usually n_l is much smaller than N_l (or in our case $K>>1$), so that one can approximate N_l-$n_l \approx N_l$. With this one gets

$$StdErr[\hat{Sum}]_{strat} = \sqrt{\sum_{l=1}^{L} N_l^2 \cdot \frac{\sigma_l^2}{n_l}} = \sqrt{\sum_{l=1}^{L} K^2 \cdot \frac{\sigma_l^2}{l}} = \sqrt{\frac{N^2}{L^2} \cdot \sum_{l=1}^{L} \sigma_l^2} = N \cdot \sqrt{\frac{1}{L^2} \cdot \sum_{l=1}^{L} \sigma_l^2} \tag{4.44}$$

The achievable accuracy of 1-in-K sampling for the special case N_f=N depends on the following sampling parameters

- the population size N
- the number of strata L (which defines the sample size n=L and the number of packets per subinterval K=N/L)

and on the spreading of packet sizes over the measurement interval, expressed by

- the variances $\sigma_1^2, \sigma_2^2, \ldots, \sigma_L^2$ of the packet sizes per stratum

As for the n-out-of-N sampling, the standard error increases, if there are more packets in the measurement interval (N increases). Since from each subinterval exactly one packet is selected the overall sample size n is equal to the number of subintervals L. Therefore the standard error also increases if there are less subintervals L, and with this less samples for the

measurement interval. Furthermore, the standard error increases, if the sum of the variances σ_l^2 of packet sizes within the subintervals increases.

4.8.1.3 General Case ($N_f<N$)

If multiple flows are active in the measurement interval the problem becomes much more complex. In 4.8.1.2 one only had to consider how the stratification affects the selection of different packet sizes. With multiple flows one additionally needs to take into account how the stratification influences the selection of packets from different flows. So, as for the systematic sampling one needs to consider both packet attributes: the packet size and the flow to which the packet belongs (flow ID).

For n-out-of-N sampling the number of packets in the sample that belong to flow *f* varies with each sample run. In section 4.6 it was shown that the number n_f of packets from flow *f* in the sample can be expressed by a binomial distributed random variable with *n* trials and a probability of success $P_f=N_f/N$. With this knowledge about the distribution of n_f it was possible to derive a formula for the expected accuracy.

For 1-in-K sampling the case is different. One randomly selects 1 packet per subinterval. The sample size *k* within the subinterval is always 1. The number k_f of packets from flow *f* within this sample can be 0 or 1. The probability that k_f is 1 (i.e., the selected packet belongs to flow *f*) depends on the total amount of packets from flow *f* in the subinterval K_f. Therefore k_f can be considered as a Bernoulli distributed random variable with a probability of success $p_f=K_f/K$.

For one sample run the number n_f of all packets from flow *f* in the sample over the whole MI can be calculated as the sum of the k_f from each subinterval.

$$n_f = k_{f,1} + k_{f,2} + \ldots + k_{f,L} \tag{4.45}$$

To calculate the accuracy per flow it is important to get the expectation and the variance of n_f. (see 4.6) For n-out-of-N sampling the distribution of n_f only depends on the proportion $P_f=N_f/N$ of packets from the flow in the whole measurement interval. It is independent of the spreading of packets from that flow over the measurement interval. For 1-in-K sampling the distribution of n_f depends on the success probabilities per subinterval *l*. The probability that a packet from flow f is selected in a subinterval is given by the proportion of packets from flow *f* within the subinterval

$$Prob(k_{f,1}=1) = \frac{K_{f,l}}{K} \tag{4.46}$$

$K_{f,l}$ denotes the number of packets from flow *f* in subinterval *l* and highly depends on the spreading of packets from flow *f* over the measurement interval. An extreme example is the case where all packets from flow *1* belong to the first subinterval. Lets consider that flow *1* has 5 packets ($N_f=5$). All packets occur in the same subinterval of size $K=5$. Now $n=5$ packets are selected from the measurement interval. If the selection is done with n-out-of-N sampling one can get $n_f= 0,1,2,3,4$ or all *5* of the 5 packets of flow *1* in the sample. With 1-in-

K sampling the number of selected packets n_f from flow 1 will always be 1 (if K=5). Figure 4-8 illustrates the example.

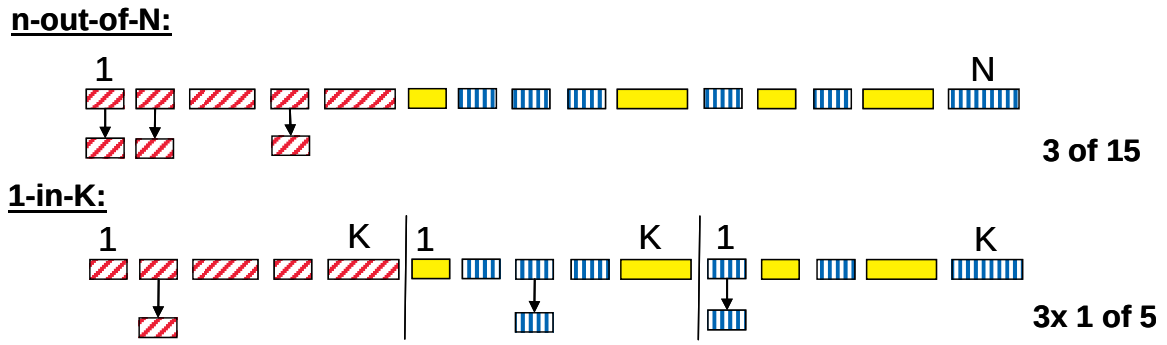


Figure 4-8: n-out-of-N vs. 1-in-K sampling

The success probabilities $Prob(k_{f,1}=1)$ per subinterval are usually unknown and can differ for each k_f. n_f is the sum of all k_f and can be modeled as multiple Bernoulli experiments with different probabilities. So for the 1-in-K case the distribution of n_f is usually not binomial, except for the case when all $k_{f,l}$ have the same probability of success. For 1-in-K sampling the following sampling parameters need to be considered:

- the population size N
- the number of strata L (which defines the sample size n=L and the number of packets per subinterval K=N/L)

Furthermore the following flow characteristics are relevant:

- the variances $\sigma^2_{f,1}, \sigma^2_{f,2}, \ldots, \sigma^2_{f,L}$ of the packet sizes of flow f per stratum (spreading of packet sizes)
- the numbers $K_{f,1}, K_{f,2}, \ldots, K_{f,L}$ of packets from flow f per stratum (spreading of flow IDs)

If the distribution of the random variable n_f is unknown one cannot provide a general formula for the accuracy as done for the n-out-of-N selection. The accuracy per flow cannot be calculated without knowledge about the spreading of flow IDs and packet sizes over the subintervals. Furthermore, parameters of the subinterval ($K_{f,l}$ or $\sigma^2_{f,l}$) cannot be estimated from the sample, because only one packet is selected per subinterval, which is not sufficient to provide a realistic estimation.

4.8.1.4 Comparison of stratified (1-in-K) to random n-of-N

One gets a better accuracy with 1-in-K sampling if the standard error for 1-in-K sampling is smaller than the standard error for n-out-of-N sampling, so if the following condition holds:

$$StdErr[\hat{Sum}]_{strat} < StdErr[\hat{Sum}]_{rand} \qquad (4.47)$$

For the single flow case a gain can be achieved if:

$$N \cdot \sqrt{\frac{1}{L^2} \cdot \sum_{l=1}^{L} \sigma_l^2} < N \cdot \sqrt{\frac{\sigma_x^2}{n}} \tag{4.48}$$

$$\frac{1}{L^2} \cdot \sum_{l=1}^{L} \sigma_l^2 < \frac{\sigma_x^2}{n} \tag{4.49}$$

Since $n=L$, this can be simplified to.

$$\frac{1}{L} \cdot \sum_{l=1}^{L} \sigma_l^2 < \sigma_x^2 \tag{4.50}$$

That means one gets a higher accuracy with 1-in-K sampling if the mean of the variances per subinterval (over all subintervals) is smaller than the variance within the whole measurement interval. The mean of the variances in the subintervals gets smaller if the individual variances are small, that means if the packet sizes occur in groups within the trace so that the sizes are more homogeneous in the subintervals than in the whole measurement interval.

If the variances σ_l^2 in all subintervals were equal to the variance σ_x^2 in the whole measurement interval one would get the same standard error (and with this the same accuracy) for 1-in-K sampling as for n-out-of-N sampling.

$$StdErr[\hat{Sum}]_{strat} = N \cdot \sqrt{\frac{1}{L^2} \cdot \sum_{l=1}^{L} \sigma_l^2} = N \cdot \sqrt{\frac{1}{L^2} \cdot \sum_{l=1}^{L} \sigma_x^2} = N \cdot \sqrt{\frac{1}{L^2} \cdot L \cdot \sigma_x^2} = N \cdot \sqrt{\frac{\sigma_x^2}{L}} = N \cdot \sqrt{\frac{\sigma_x^2}{n}} \tag{4.51}$$

In this case no gain can be achieved.

The problem is that one needs information about the variances of the packet sizes within the subintervals in order to be able to predict the accuracy for the 1-in-K method. In contrast to the variance of packet sizes within the measurement interval (needed for the n-out-of-N case), the variance per subinterval cannot be estimated from the sample. Only one packet is selected per subinterval and a sample size of $k=1$ is definitively too small to estimate a variance for the subinterval.

Knowledge about the correlation between packet sizes and their positions in the measurement interval would help to approximate the stratification gain. But it is quite unlikely that one has a-priori knowledge about this. An investigation of the correlation would require the analysis of much more packets and therefore would drastically reduce the resource savings achieved with sampling.

For the multi-flow case additionally the spreading of flow IDs has to be considered.

If packets from a flow occur in packet trains (i.e., subsequent packets from the flow with no or only few packets from other flows in between) 1-in-K sampling may ensure that at least some packets of this flow are selected. On the other hand it s less likely that many packets of the same flow are selected.

So if one has many small flows that occur in packet trains it is more likely that for each flow at least one packet is selected (i.e., one detects most of the existing flows) but the number of selected packets per flow is small (i.e., the estimation accuracies per flow are small).

4.9 Prediction of the Estimation Accuracy and Parameter Adaptation

Providing information about the expected estimation accuracy is very important when sampling methods are deployed for usage-based accounting. Only with such information customers are able to assess the degree of potential estimation errors.

Nevertheless, as shown above, the estimation accuracy depends not only on sampling parameters (like sampling rate *n* and measurement interval length *N*) but also on traffic characteristics like mean and standard deviation of packet sizes or number of packets per flow. These traffic parameters are unknown and usually vary over different measurement intervals.

Therefore it is now investigated how the unknown variables can be approximated in order to calculate an approximated standard error without previous knowledge about the trace. When working with an approximated standard error, one can get confidence limits that differ from the theoretical values. The size of that difference depends on the accuracy of the approximation.

Table 4-8 summarizes from which traffic parameter the estimation accuracy depends for the different cases with n-out-of-N sampling. The unknown variables in the formulas above are the following: the mean packet size μ_{x_f} in the flow, the variance $\sigma^2_{x_f}$ of the packets sizes in the flow and the number N_f of packets that belong to flow *f*. So in order to calculate the estimation accuracy, one needs to approximate these unknown values.

		Case A, N_f=N	**Case A, N_f<N**	**Case B, N_f=N**	**Case B, N_f<N**
Sampling Parameters	n	Preconfigured	Preconfigured	Preconfigured	Preconfigured
	N	Preconfigured	Preconfigured	Preconfigured	Preconfigured
	n_f	n=nf → preconfigured	Preconfigured (per flow)	Known after sampling	Known after sampling
Traffic Characteristics	N_f	Nf=N → Preconfigured	Known before sampling	Nf=N → Preconfigured	Unknown
	σ_{x_f}	Unknown	Unknown	Unknown	Unknown
	μ_{x_f}	Not relevant	Not relevant	Unknown	Unknown

Table 4-8: Dependency of Estimation Accuracy from Sampling Parameters and Traffic Characteristics for n-out-of-N Sampling

An approximation of the traffic characteristics can be done by different methods:

- ***Theoretical considerations:*** One can approximate the values by applying theoretical considerations. Examples are calculating boundaries for the values under consideration of maximum and minimum possible packet sizes or assuming worst case values for certain scenarios. The incorporation of a priori information (e.g., about active applications and their typical packet sizes) and assumptions from experience (e.g., assumptions about the packet size distribution) allow increasing the precision of the approximation.
- ***Estimation:*** One can estimate the unknown values from the sampled values of the actual measurement interval. One can calculate certain characteristics of the sample (e.g., mean, variance of the packet sizes in sample) and derive estimates for the population. The sampled values of the actual measurement interval are required for the calculation. That means one can only calculate the estimation accuracy ***after*** the sampling process for this measurement interval has completed.
- ***Prediction:*** One can approximate unknown values by looking at their values at previous points in time. For instance one can use measurement results from previous measurement intervals to predict the values for the actual measurement interval. The prediction of these values has the advantage, that it allows a calculation of the accuracy in advance. This is a very desirable feature. It allows a direct reaction on expected values and therefore provides the basis for an adaptation of sampling parameters for maintaining a predefined accuracy level. Therefore prediction allows the use of adaptive sampling techniques. The difficulties here are to find a good model for the prediction of the different parameters, to find the right time scales for the prediction and to handle limitations for changing parameters dynamically (see 3.3.8).

4.9.1 Approximation with Theoretical Considerations

In order to provide theoretical boundaries on the mean value and the variance of the packet sizes in the flow, the minimum expected packet size b_{min} and the maximum expected packet size b_{max} in the network are needed. With this one can make the following statements about the mean μ_{x_f} and the variance $\sigma^2_{x_f}$ of the packets of flow *f*: Obviously the mean value lies between b_{min} and b_{max}.

$$b_{min} \le \mu_{x_f} \le b_{max} \tag{4.52}$$

One gets the highest variance for an equal distribution of packet sizes. With this worst case distribution the variance get the value $\frac{1}{4}b^2_{\max}$. Therefore the variance has the following range.

$$0 \le \sigma^2_{x_f} \le \frac{1}{4}b^2_{\max} \tag{4.53}$$

A variance of 0 occurs if the flow consists of packets with equal sizes. In this case all $x_{i,f}$ are equal to the mean value μ_f. The highest variance occurs, if only packets with the smallest and the largest possible size are in the flow and they occur with the same proportion (50% large

and 50% small packets). The proportion P_f of packets from flow *f* in the population lies between 0 and 1.

$$0 \le P_f \le 1 \tag{4.54}$$

In the formula for the variance for n-out-of-N sampling, case B:

$$V[\hat{S}um_f] = \frac{N^2}{n} \cdot \left(\sigma_{x_f}^2 \cdot P_f + \mu_{x_f}^2 \cdot P_f \cdot (1 - P_f) \right) \tag{4.55}$$

two terms depend on P_f. The first term is linear dependent and the second term depends on $P_f \cdot (1 - P_f)$. The first term gets its maximum at $P_f = 1$. The second term gets its maximum if at $P_f = 0.5$.

$$0 \le P_f \cdot (1 - P_f) \le 0.25 \tag{4.56}$$

If the highest values for mean and variance are used in (4.55) one gets:

$$V[\hat{S}um_f]_{\max} = \frac{N^2}{n} \cdot \left(\frac{1}{4} \cdot b_{\max}^2 \cdot P_f + \frac{1}{4} \cdot b_{\max}^2 \cdot P_f \cdot (1 - P_f) \right) = \frac{N^2}{4 \cdot n} \cdot b_{\max}^2 \left(2 \cdot P_f - P_f^2 \right) \tag{4.57}$$

The maximum of this variance is at $P_f = 1$.

$$V[\hat{S}um_f]_{\max} = \frac{N^2}{4 \cdot n} \cdot b_{\max}^2 \tag{4.58}$$

This is the worst case variance for n-out-of-N sampling, case B, if no further information is available. Please note that this value gives an upper bound and therefore the variance can get quite high. So if possible one should take further information into account to approximate the estimation accuracy.

4.9.2 Estimation from Actual Sampled Values

The mean value μ_f of the packet sizes in flow *f* in the actual measurement interval can be estimated by the mean value of the packet sizes in flow *f* in the sample.

$$\overline{x}_f = \frac{1}{n_f} \cdot \sum_{i=1}^{n} x_{i,f} \tag{4.59}$$

The variance of the packet sizes in the parent population can also be estimated from the sampled values

$$s_{x_f}^2 = \frac{1}{n_f - 1} \cdot \sum_{i=1}^{n_f} (x_{i,f} - \overline{x}_f)^2 \tag{4.60}$$

Please note that the calculation is done with $n_f - 1$ instead of just using the variance of the sample (with divisor n_f). Only with this one gets an unbiased estimate of the variance. The number N_f of packets of flow *f* in the population can be approximated based on the proportion of packets of flow *f* in the sample

$$N_f \approx \hat{N}_f = \frac{n_f}{n} \cdot N \tag{4.61}$$

$$P_f \approx \frac{n_f}{n} \tag{4.62}$$

If those values are inserted in (4.55) one gets:

$$V[\hat{Sum}_f]_{approx} = \frac{N^2}{n} \cdot \left(\left(s_{x_f}^2 + \overline{x}_{x_f}^2 \right) \cdot \frac{n_f}{n} - \overline{x}_{x_f}^2 \cdot \frac{n_f^2}{n^2} \right) \tag{4.63}$$

With this formula the estimation accuracy can be approximated from the sampled values ***after*** the sampling process has completed. Since the values for mean, variance, and packet proportion of the sample provide a more precise estimate of these values than simply assuming the maximum values, one gets a more precise approximation of the estimation accuracy than just with theoretical considerations.

4.9.2.1 Recommendation for Cisco NetFlow: Storage of the Square Sum

Cisco NetFlow stores and updates only aggregated statistics per flow. The individual per packet information (packet size) is not stored. Therefore one cannot derive arbitrary metrics from the sampled packets.

Since the number of sampled packets and the sum of bytes of all sampled packets is stored, it is possible to calculate the mean packet size of the sampled packets. Nevertheless, another very important parameter of the packet size distribution, the variance, cannot be calculated since the original per packet information is lost. From the estimated mean one can derive an estimate for the flow volume. But one cannot approximate the standard error because the formula requires estimates for number of packets, packet size mean and variance.

A possible solution to allow an estimation of the variance is to store and continuously update not only the sum of the sampled packets but also the square sum of the sampled packets. This is a common technique used e.g., in calculators. The formula for calculating the variance from the sum and the square sum is shown below:

$$V[X] = E\left[X^2\right] - E[X]^2 = \frac{1}{\mathrm{n}} \cdot \sum_i x_i^2 - \frac{1}{\mathrm{n}^2} \cdot \left(\sum_i x_i \right)^2 \tag{4.64}$$

with $E[X] := \sum_i x_i \cdot P_i = \frac{1}{\mathrm{n}} \sum_i x_i$ and $E[X^2] := \sum_i x_i^2 \cdot P_i = \frac{1}{\mathrm{n}} \sum_i x_i^2$.

Therefore a strong recommendation for improving Cisco NetFlow is to store (and update) besides the sum of bytes of the observed or sampled packets of a flow also the square sum of bytes of the observed or sampled packets of a flow.

The simple storage of this additional value allows the derivation of much more information about the flow characteristics not only if sampling is applied. In case that all observed packets are investigated storing the square sum allows the calculation of the variance of the packet sizes in the flow. In case that sampling is done it allows the calculation of the variance of the packets sizes in the sample. This variance can be used to provide an estimate for the variance of all packets in the flow. With this estimate a more accurate approximation of the estimation

accuracy is possible and one can more precisely calculate the boundaries of the confidence interval.

4.9.3 Prediction from Previous Samples

If one wants to predict the accuracy before the sampling process, one has to predict the traffic characteristics from measurements of previous measurement intervals. If sampling is deployed one gets only the sampled values as basis for this prediction. Therefore the prediction has to be based on estimates of the traffic characteristics.

The prediction of the characteristics of the actual measurement interval can be based on the estimate of the directly preceding measurement interval only or on multiple previous measurement intervals. E.g., one could estimate the mean packet sizes of a flow f as follows:

$$\mu_i' = f\left(\hat{\mu}_{i-1}\right) \tag{4.65}$$

$$\mu_i' = f\left(\hat{\mu}_{i-1}, \hat{\mu}_{i-2}, \hat{\mu}_{i-3}, \ldots,\right) \tag{4.66}$$

With this there are two errors that need to be considered. First, an estimation error is made, when estimating μ_{i-1} by $\hat{\mu}_{i-1}$ from the samples in measurement interval i-1. Second, one gets a prediction error when μ_i' is predicted from $\hat{\mu}_{i-1}$.

A similar problem is addressed in [ChPZ02] for measuring traffic load by using an autoregressive (AR) model. In that approach the predicted estimation accuracy is used to adapt sampling parameters in order to keep the estimation accuracy constant. Although the approach is quite valuable, one has to consider both errors (from prediction and estimation). Furthermore, the adaptation of sampling parameters requires a dynamic control of the measurement configuration.

The usability of this method highly depends on reliable methods to predict the actual proportion from previous measurement intervals. It is questionable whether the relevant traffic characteristics are stationary enough too allow a good prediction.

4.9.4 Adaptive Sampling

In the sections above it was shown that the estimation accuracy depends not only on the sampling parameters but also on traffic characteristics. That means if the traffic characteristics change, one gets different estimation accuracies. It is quite likely that one gets different traffic characteristics for different measurement intervals. Therefore one would get a different accuracy per measurement interval.

Adaptive sampling provides an approach to maintain a more or less stable estimation accuracy by adapting the sampling parameters to the changing traffic characteristics. If for instance the variance of packet sizes increases, the estimation accuracy would decrease if the sampling parameters remain static. But one could maintain the accuracy if one increases the sample size.

One problem is that one needs to know or at least predict the changes in the traffic characteristics before they occur (see 4.9.3), in order to be able to adjust the sampling parameters in time. This is not trivial. Another problem is that the re-configuration of sampling parameters has to be fast and usually cause some configuration overhead. So, in order to assess the applicability of adaptive sampling one needs to investigate

- How fast and how dynamic do the relevant traffic characteristics change
- How good can one predict changes in traffic characteristics (e.g., from previous samples)
- How fast can sampling parameters be adapted

From (4.27) one could see that for random n-out-of-N sampling, the following three traffic characteristics are relevant:

- the number N_f of packets from flow f in the parent population
- the standard deviation σ_{x_f} of the packet sizes in the flow
- the mean μ_{x_f} of the packet sizes in the flow

The sampling parameters that can be configure and therefore adapt to the situation are the following:

- Measurement interval N
- Sample size n

In order to investigate whether adaptive sampling could be applied, one would need to investigate how fast and dynamically the relevant traffic characteristics change over time.

5 Experiments for Flow Volume Estimation

In this chapter the experiments for the flow volume estimation are described and experimental results are analyzed.

5.1 Questions for Empirical Investigations

In chapter 4 it was shown that traffic characteristics are important input parameters for the achievable accuracy. Here the relevant traffic characteristics from real traffic traces are investigated in order to find out what values those parameters have in reality. With this it can be assessed in which range the accuracy usually lies.

A second outcome of the empirical investigations is the validation of the model. For this it is checked whether the empirical results conform to the predicted behavior from the model. Furthermore, several assumptions were made in order to derive models. It needs to be checked whether those assumptions hold true in reality and how the real accuracy evolves if assumptions are violated. A third point is the question whether the accuracy can be approximated in reality, where only the information about samples is available.

The following questions should be answered by empirical investigations.

- Investigation of traffic characteristics (input to mathematical models):
  - Number of packets and number of bytes per flow (flow volumes)
  - Packet size distributions: packet size mean and variances per flow
- Theoretically expected vs. empirical estimation accuracy
  - Evaluation of the model: Does the empirical accuracy differ from the theoretical accuracy predicted by the model?
  - Achievable accuracy when model assumptions do not hold: How does the empirical accuracy develops when model assumptions do not hold?
  - Empirical accuracy for stratified schemes: What empirical accuracy is achieved with stratified schemes?
- Approximation of estimation accuracy from samples
  - How good can the estimation accuracy be approximated from the samples?

This chapter describes the experiments that were performed to answer those questions and comments the results. A complete trace analysis, with all measured packets, is performed to find out the traffic characteristics of the flows in the trace. From this the theoretical accuracy for n-out-of-N can be calculated with the model. Then multiple sampling runs are simulated with different sampling algorithms and sampling parameters to compare theoretical and empirical accuracy for different cases.

5.2 Software for Trace Analysis and Sampling Simulation

Trace analysis and sampling simulation is done by C and C++ programs, statistical analysis and graphical representation of results is done with the statistic software R and partly with Excel. A software package developed in the VEGAS project is used for the trace analysis and the simulation of different sampling methods [VEGAS-SW].

The tool takes trace files in packet file format as input. The packet file format is a common binary format for storing trace files, which is also used by the tool *tcpdump*. The name of the trace file, sampling parameters and other configuration data is specified in a configuration file. The functions to perform the sampling methods are collected in a library and can also be used by other programs. The software can perform a complete analysis of the trace file and simulate multiple sample runs in accordance to the configuration data. The output is written into a MySQL database.

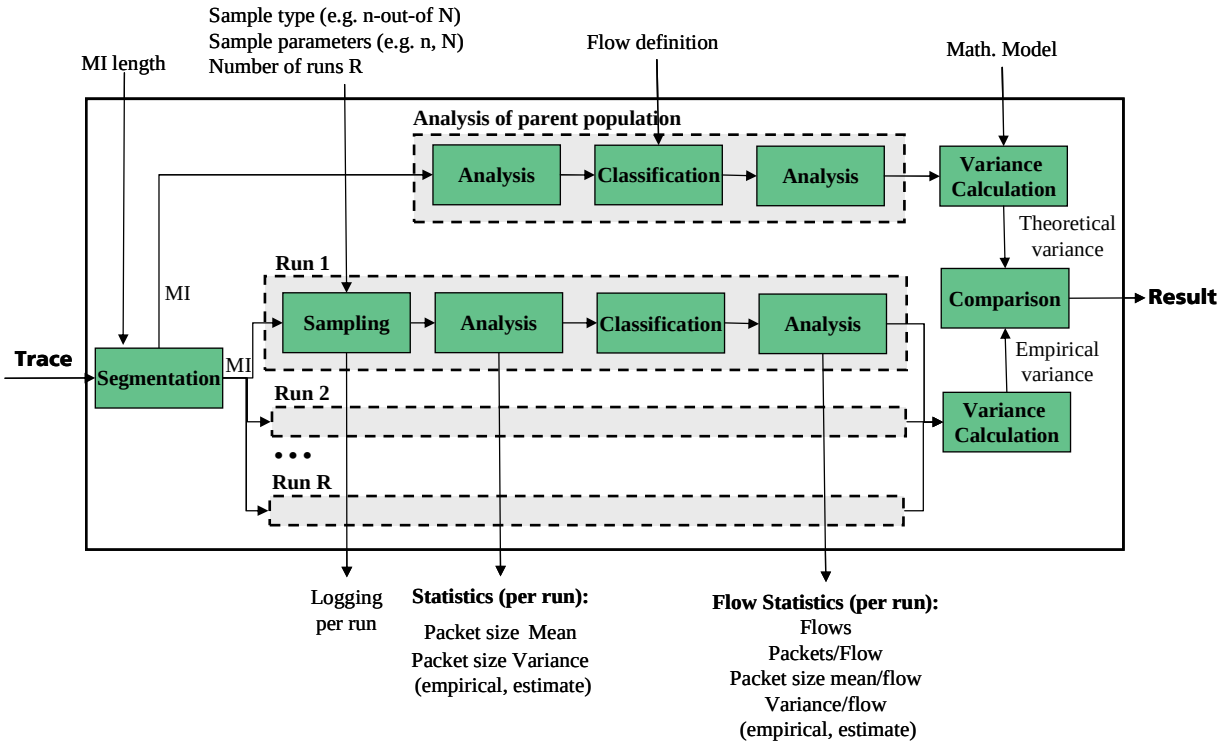


Figure 5-1: Software for Trace Analysis and Sampling Simulation

The program first splits the traces into measurement intervals in accordance to a measurement interval length specified in the configuration file. It classifies all packets into flows with respect to the classification rules specified in the configuration file. Then it performs a complete analysis of the trace in order to get the overall volume and the real volume of each flow. It also calculates further flow characteristics, like packet size mean and variances, needed for the calculation of the theoretical estimation accuracy per flow.

The program then performs multiple sampling runs over each measurement interval. For each sampling run an estimate for the volume for each flow is computed. Furthermore estimates of the relevant flow characteristics are calculated from the selected packets for each run.

All results for all flows and runs are stored in a MySQL database. An analysis program calculates the theoretical expectation and standard error from the traffic parameters by using the theoretical model. The empirical expectation and standard error are derived from the estimates of all runs.

5.2.1.1 Implementation of Sampling Algorithms

The functions for the implementation of sampling methods are collected in a library. For statistical functions the GNU Scientific Library (GSL) [GSL] is used. The random number generator "Mersenne Twister" of Makoto Matsumoto and Takuji Nishimura [MaNi98] is used for the generation of random numbers.

5.2.1.1.1 n-out-of-N Sampling

For n-out-of-N sampling exactly n_T different random numbers in the range *[1, N]* need to be generated. In order to ensure that one does not get repeating random numbers a bitfield of size N is used and initialized with the value zero. At the beginning of each measurement interval n_T uniformly distributed integer random numbers are drawn from the range *[1, N]*. For each selected number the corresponding bit in the bitfield is set to one. If the corresponding bit has already been set to 1 by a previous drawing, the current random number is discarded and a new one is generated. Then all packets from the measurement interval are selected with packet positions that corresponds to the bits set to one in the bitfield.

5.2.1.1.2 Probabilistic Sampling

For probabilistic sampling the selection probability p is calculated by dividing the target sample size n_T by the measurement interval length N ($p=n_T/N$). A Bernoulli distributed random number with probability p is generated for each packet. If the random number is 1, the corresponding packet is selected. Since for probabilistic sampling the real sample size can differ from the target sample size, two options are allowed for the extrapolation. One option is to extrapolate with the target sample size n_T. This needs to be done for instance if the real sample is unknown. Another option is to use the real sample size n_R for the extrapolation.

5.2.1.1.3 Systematic Sampling

For systematic sampling every K^{th} packet is selected. A random number generation is therefore not needed. The sample interval K is calculated by dividing the measurement interval length N by the target sample size n_T ($K=N/n_T$). Since K has to be an integer it is rounded off if N is no multiple of n. With this one gets a higher real sample fraction if N is no multiple of n. For instance for a target sample fraction of 40%, and a measurement interval of $N=10,000$ one would need to select $n=4,000$ packets. Since $K=10,000/4,000=2.5$, the value is rounded off to $K=2$. That means every second packet is selected and one gets a real sample size of $n_R=5,000$ (real sample fraction = 50%). The extrapolation is always done with the real

sample fraction n_R. In order to get different sets of samples for different sample runs, the starting point for the selection (first selected packet) is chosen randomly in the experiments.

5.2.1.1.4 1-inK Sampling

For 1-in-K sampling the measurement interval is divided into subintervals. Each subinterval contains $K=N/n$ packets. If the measurement interval length N is no multiple of n, i.e., K is no integer, the K is round off to the next smaller integer. A uniformly distributed integer random from the range $[1, K]$ is generated for each subinterval. Then the packet with the corresponding position in the subinterval is selected.

5.2.1.2 Calculation of Empirical Bias and Precision

The absolute and relative empirical bias is calculated from the sum of the estimated flow volumes $\hat{Sum}_{f,r}$ from all R sampling runs as follows:

$$Bias_{f,abs,emp} = E\left[\hat{Sum}_f\right] - Sum_f = \left(\frac{1}{R}\cdot\sum_{r=1}^{R}\hat{Sum}_{f,r}\right) - Sum_f \tag{5.1}$$

$$Bias_{f,rel,emp} = \frac{\left(\frac{1}{R}\cdot\sum_{r=1}^{R}\hat{Sum}_{f,r}\right) - Sum_f}{Sum_f} \tag{5.2}$$

The absolute and relative empirical standard error, which is used as a measure for the precision, is calculated from the sum and the squaresum of the estimated flow volumes $\hat{Sum}_{f,r}$ from all R sampling runs as follows:

$$StdErr_{f,abs,emp}\left[\hat{Sum}_f\right] = \sqrt{E\left[\hat{Sum}_f^2\right] - E\left[\hat{Sum}_f\right]^2} = \sqrt{\frac{1}{R}\cdot\sum_{r=1}^{R}\hat{Sum}_{f,r}^2 - \left(\frac{1}{R}\cdot\sum_{r=1}^{R}\hat{Sum}_{f,r}\right)^2} \tag{5.3}$$

$$StdErr_{f,rel,emp}\left[\hat{Sum}_f\right] = \frac{StdErr_{f,abs,emp}\left[\hat{Sum}_f\right]}{Sum_f} \tag{5.4}$$

5.3 Traces

Experiments were performed with different traces. For the VEGAS project large traces from a major European telecom operator and from a European network provider that interconnects multiple universities where available. Some of the traces cover multiple days and where collected at different times. Furthermore, a 12 hour trace from the Waikato Internet Traffic Storage [WITS] measured by the WAND group [WAND] at a New Zealand Internet exchange point was used for the analysis and sampling simulation. Since the VEGAS traces are not public, here only the results for the 12 hour trace from the New Zealand exchange point are shown (NZIX trace).

5.4 Analysis of Traffic Characteristics

The trace used for the sampling simulation contains 12 hours continuously measured data. Measurement setup and an initial analysis of protocols and applications in use can be found at [WITS].

First the traffic characteristics of the complete trace are analyzed. For this the trace is split into measurement intervals and all packets in the trace are classified into flows. With a measurement interval length of 1,000,000 packets, 65 complete measurement intervals can be formed. Remaining packets that are too few to fill a complete measurement interval are neglected.

Two different classification schemes were used. The first one (S24D24) distinguishes flows with respect to source and destination network both with a 24 bit netmask. The second one (S24D00) is a more coarse classification that distinguishes flows only with respect to the source network with a 24 bit netmask. That means all packets that originate from the same source network, specified by the first 24 bits in the address belong to the same flow. All packets were classified into flows in accordance to these classification rules. If packets with the same flow ID are observed in different measurement intervals they are counted as separate flows. Therefore one gets more flows if the analysis is done after the trace is split into MIs. Table 5-1 summarizes the parameter settings.

Parameter	**Value**
Trace	NZIX1 (20000706_120000.stripped.pf)
MI length	1,000,000 (65,000,000 for complete trace)
Classification S24D00	Source IP address with netmask 0xffffff00
Classification S24D24	Source and destination IP address with netmask 0xffffff00

Table 5-1: Parameter Settings

The S24D24 classification results in 76,560 different flows in the whole trace. If measurement interval boundaries are considered flows are split and one gets 537,138 flows in the trace. With the S24D00 classification there are only 1,486 different flows if measurement interval boundaries are not considered. If the measurement interval boundaries are considered the trace contains 79,383 flows. Since this value is higher than the number of flows without measurement interval separation, several flows span multiple measurement intervals. Since measurement results are reported per measurement interval, in the following trace analysis it is assumed that flows are separated by the measurement boundaries. In the following only results for the more coarse grained classification (S24D00) are shown, if not stated otherwise.

5.4.1 Flow Characteristics

In accordance to the model introduced in 4.6, the estimation accuracy of the volume per flow depends on the number of packets for this flow, on the mean packet size and on the packet

size variance. In the following the relevant flow characteristics are investigated for all flows in the trace for a classification in accordance to the source network (S24D00).

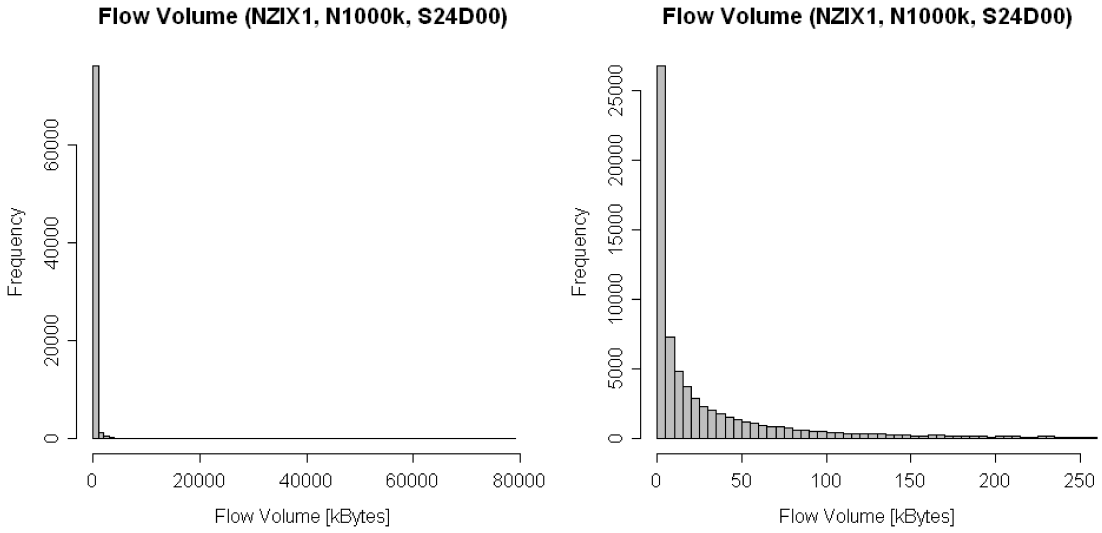


Figure 5-2: Flow Volume (Left: all Flows. Right: Small Flows)

Figure 5-2 shows the histogram of the flow volume. The left histogram contains all flows. The right histogram shows only flows with a volume below 250 kbytes. The majority of flows are small. The mean flow volume is 253 kbytes and the median lies at 15 kbytes. The smallest flow has a volume of 29 bytes. The volume of the largest flow is 78 Mbytes.

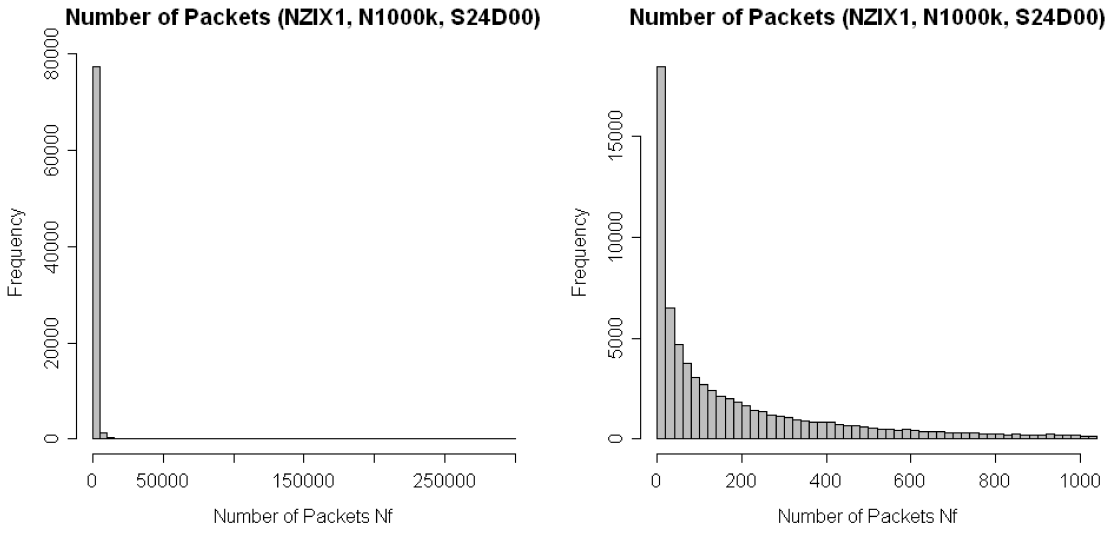


Figure 5-3: Number of Packets (Left: all Flows. Right: Small Flows)

Figure 5-3 shows the histogram of the number of packets per flow. The left histogram contains all flows. The right histogram shows only flows with less than 1000 packets. Also with regard to number of packets, the vast majority of flows are small. The mean number of packets is 818.8. The median lies at 125 packets. That means 50% of all flows in the trace consist of 125 or less packets. The smallest flows consist of only 1 packet. The largest flow contains 296,403 packets.

The vast majority of flows are small in volume as well as in number of packets. Furthermore there is a big gap between the few large flows and the majority of small flows. Similar observations on the distribution of flow sizes for backbone traffic have also been made by others (see 2.1.2). It is important to notice that with the given classification only few flows in the trace conform to the initial assumption (4.12) in chapter 4.6.1. This assumption about the packet proportion (P_f>0.1) was a precondition to derive a model for case B. The difference of empirical results to the model are investigated in section 5.5.1.

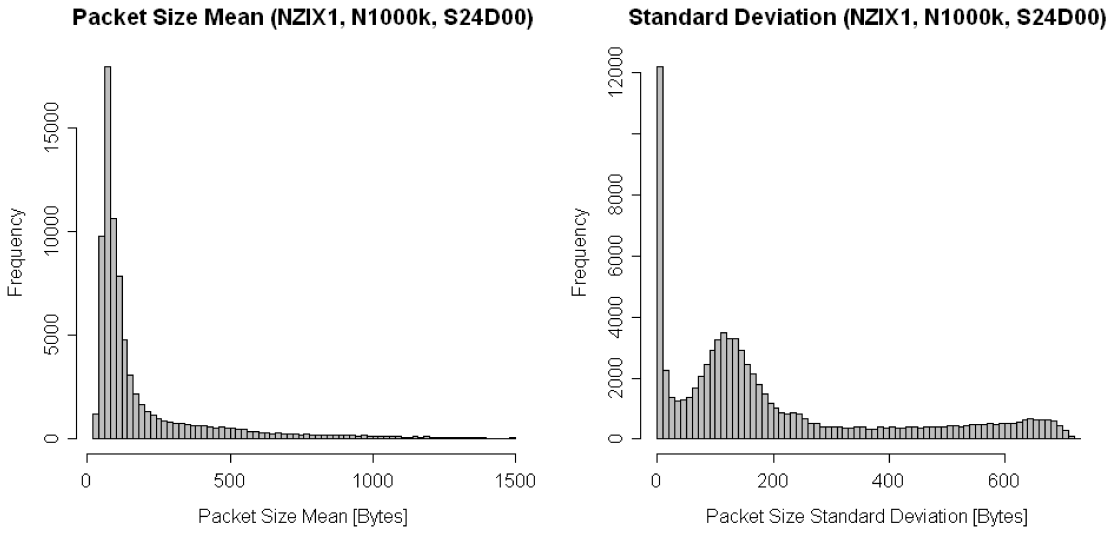


Figure 5-4: Mean and Standard Deviation of Packet Sizes

Figure 5-4 shows the histograms of the packet size means and standard deviations for all flows. For an MTU of 1500 bytes, the theoretical range for the mean packet size is from 0 to 1500 bytes (see formula (4.52) in section 4.9.1). In the trace most flows have a mean packet size around 100 bytes. Only 25% of all flows have a mean packet size larger than 200 bytes.

For an MTU of 1500 bytes, the maximum theoretical variance of the packet sizes per flow is 562,500 bytes, the maximum theoretical standard deviation is 750 bytes (see formula (4.53) in section 4.9.1). In the trace the standard deviations spread over the whole range with peaks at zero, and around 130 bytes. The peak at the standard deviation of zero is caused by flows with packets of equal sizes. In the investigated trace there are several flows that consist only of one packet. Since those flows have only one packet size, they also have a standard deviation of zero. Table 5-2 contains a summary of the flows characteristics.

	Volume [Bytes]	#Packets	Mean [Bytes]	StdDev [Bytes]
Minimum	29	1	29	0
1st Quartile	2,138	24	69	60.74
Median	15,785	125	100	132.50
3rd Quartile	72,374	417	203.3	267.60
Maximum	78,287,277	296,403	1,500	729.20
Mean	253,035	818.80	203.50	200.50
StdDev	1,844,059.23	6,944.33	248.44	198.94

Table 5-2: Summary of Flows Characteristics

Figure 5-5 shows a summarized representation of all 79,383 flows in the NZIX trace. Each dot represents a flow. The dimensions are the three flow characteristics that are relevant for the estimation accuracy: The number of packets, the packet size mean and the packet size variances (here represented by the standard deviation). As already observed in the histograms above one can see that only few large flows exist and that there is a large gap between the majority of small flows and the few large flows. It also can be observed that some combinations of packet size mean and standard deviation are more likely than others.

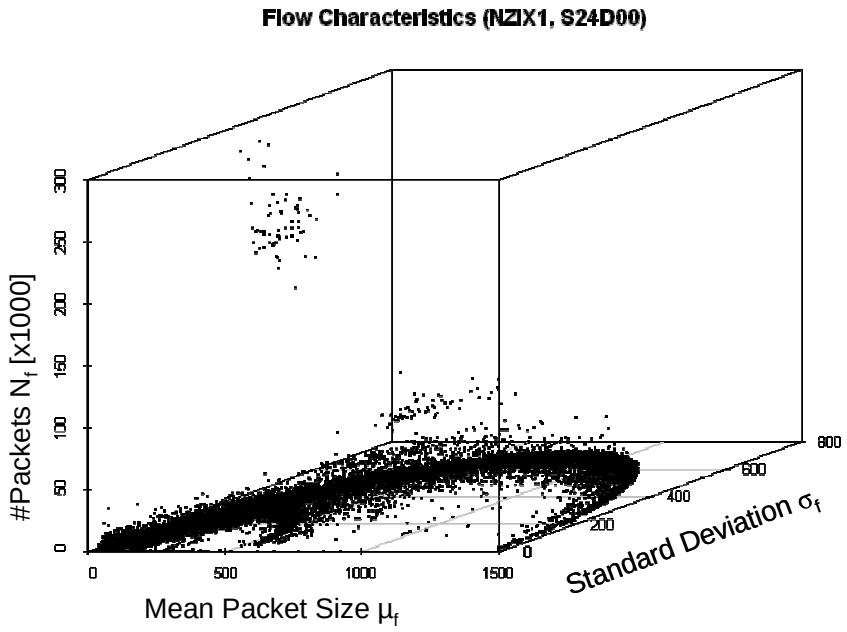


Figure 5-5: Flow Characteristics of all Flows in NZIX Trace (S24D00)

The distributions of flow characteristics over individual measurement intervals look similar to the distributions over the whole trace. Figure 5-6 shows the characteristics of all 537,138 flows in the trace for the more fine grained classification rule (S24D24). One can see that there are even less large flows. In the following analysis always the more coarse grained classification (S24D00 is used), if not stated otherwise.

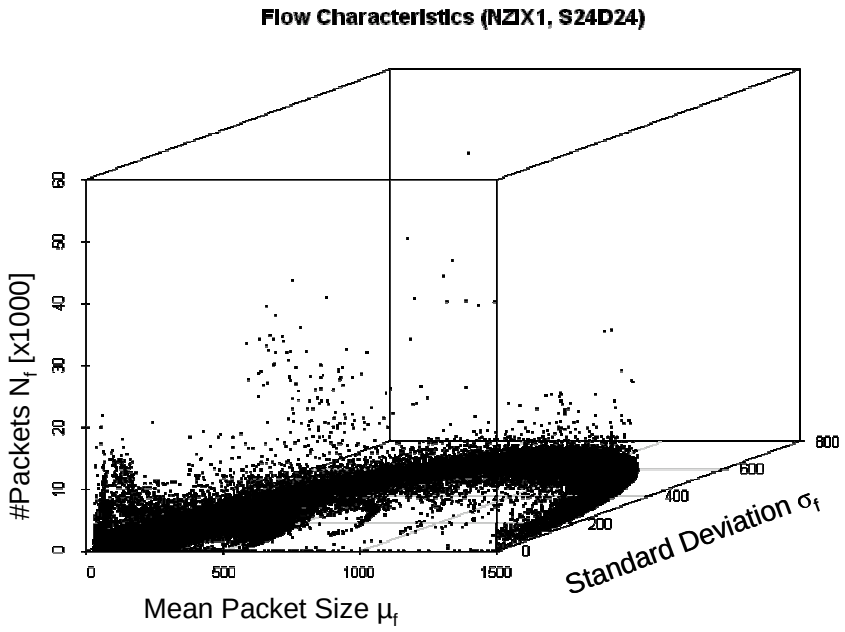


Figure 5-6: Flow Characteristics of all Flows in NZIX Trace (S24D24)

5.5 Sampling Experiments

With the results of the flow analysis and the mathematical model for n-out-of-N sampling one can derive the theoretical estimation accuracy per flow. In order to investigate what accuracy can be achieved in reality, various sampling experiments are performed. The experiments are needed in order to answer the following questions:

- Evaluation of the model (empirical vs. theoretical accuracy)
- Achievable accuracy when model assumptions do not hold
- Empirical accuracy for stratified schemes
- Approximation of estimation accuracy from samples

For this the empirical bias and standard error for different schemes and parameter settings is calculated from the sampling simulations. Different sampling methods and sample fractions were used (Table 5-3).

Parameter	Value
Trace	NZIX1 (20000706_120000.stripped.pf)
Measurement Interval	1..65
Sampling Method	nofN, 1inK, systematic
MI length N	1,000,000
Sample Fraction f	5 % -50%
Number of Runs R	1000
Classification	S24D00 (source network only)

Table 5-3: Parameter Settings

For a proper comparison with 1-in-K and systematic sampling, only sample fractions are used that result in an integer $K=N/n$. For each experiment 1000 sample runs were performed. A summary of the sampling experiments for the WAND trace for which results are presented here is given in appendix G. The bias and the standard error, derived from the empirical variance of the estimates from multiple sampling runs in the simulation, are used to assess and compare the sampling methods.

5.5.1 Comparison of Empirical Results with n-out-of-N Model

With this first investigation it is checked whether the empirical results from the n-out-of-N sampling simulation are close to those values that are expected by the model. To validate the model the results for the empirical standard errors from the experiments for n-out-of-N sampling are compared with the theoretical expected standard error for each flow. Furthermore, it is investigated how the estimation accuracy evolves compared to the model for cases where the initial model assumptions are not valid.

5.5.1.1 Bias

The theoretically expected bias for n-out-of-N sampling is zero (see (4.24) in 4.6.2.2). That means the empirical bias itself represents also the difference to the value expected by the model. Figure 5-7 shows the histograms of the absolute (left) and the relative (right) empirical bias for n-out-of-N sampling with a sample fraction of 5%.

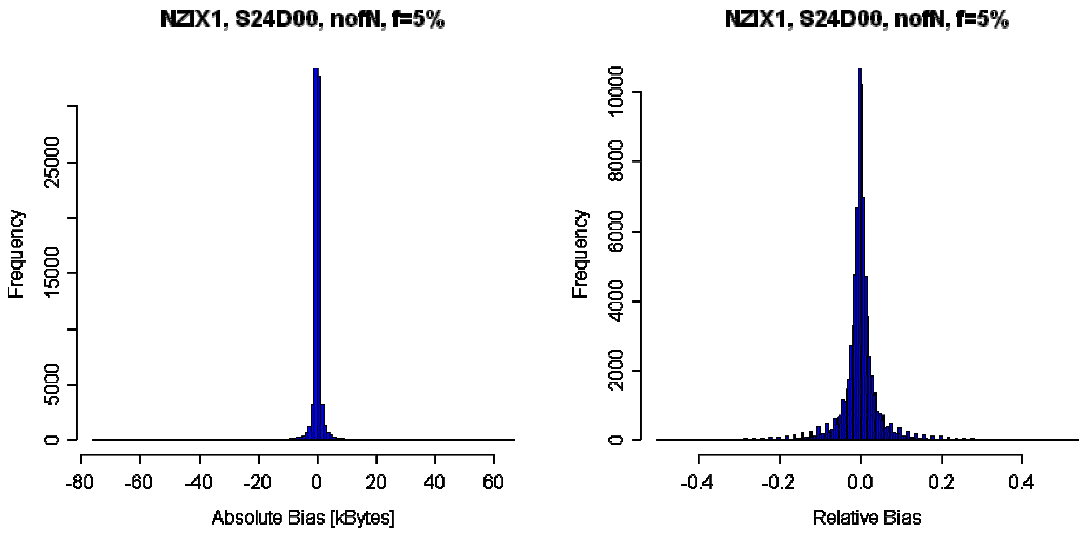


Figure 5-7: Absolute and Relative Empirical Bias for n-out-of-N, f=5%

It can be seen that for most flows the bias is zero or very close to zero. Variations occur due to the finite number of sampling runs. The bias further decreases if a larger number of runs is used. A further reason for the deviations from the model may be the presence of small flows for which the model assumptions do not hold. For larger sample fractions the bias decreases.

Figure 5-8 shows the absolute (left) and the relative (right) empirical bias over the flow proportion P_f, i.e. the number of packets per flow divided by the number of packets in the measurement interval. Since there are much more small flows, there are much more dots at lower flow proportions. If the flow proportion increases, also the flow volume increases.

But it can be seen that the absolute bias increase only a little bit. The relative bias decreases if the flow proportion increases and gets very small for larger flows. It can be seen that the bias deviations from the model are mainly caused by very small flows.

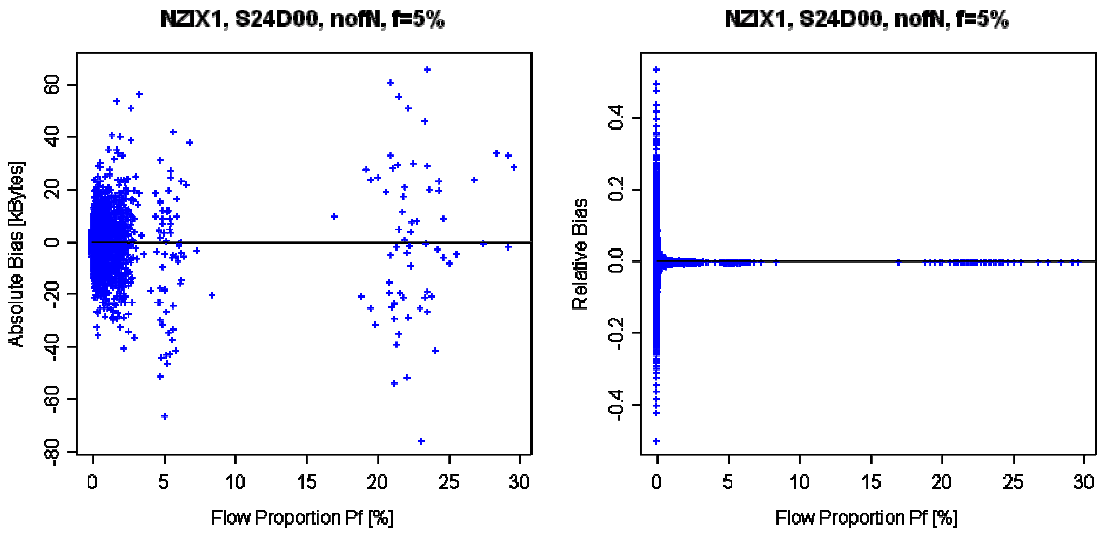


Figure 5-8: Absolute and Relative Empirical Bias for n-out-of-N over Flow Proportion, f=5%

5.5.1.2 Precision

For each flow a theoretical standard error can be calculated from the real flow characteristics and the mathematical model. The empirical standard error can be derived from the sampling experiments. The relative difference between the empirical and the theoretical relative standard error is calculated for each flow in each measurement interval.

$$diff_{rel} = \frac{StdErr_{rel,empiric} - StdErr_{rel,theoretic}}{StdErr_{rel,theoretic}} \tag{5.5}$$

A positive difference indicates a higher empirical standard error (worse accuracy) than theoretically expected. A negative difference indicates that empirical results show a higher accuracy than predicted by the model. A relative difference of 0.1 means, that the difference between empirical and theoretical standard error is 10% of the theoretical standard error. For a relative theoretical standard error of 0.5 that means that one gets an empirical error of 0.55 instead. For a relative theoretical standard error of 0.001 one would get a empirical error of 0.0011.

Figure 5-9 shows the histogram of the relative differences between empirical and theoretical standard error (left diagram) and the relative differences to the model over the flow proportion (right diagram).

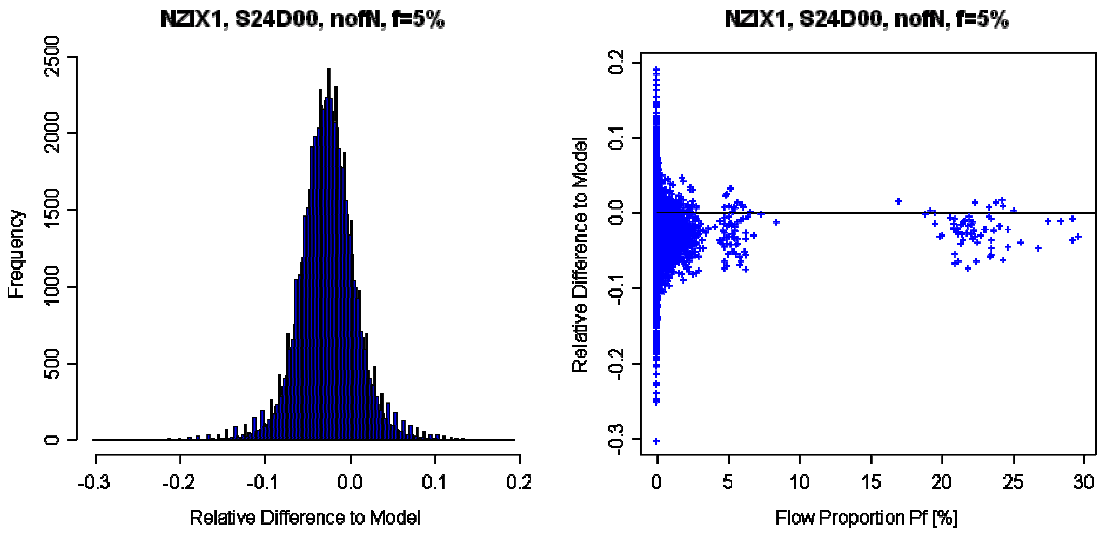


Figure 5-9: Differences between Empirical and Theoretical Standard Error for n-out-of-N, f=5%

For most flows the empirical results are close to the model (small difference). The minimum difference is -0.3 and the maximum difference is 0.19. The median (-0.026) and the third quartile (-0.007) are negative. That means in most of the cases where the achieved accuracy differs from the model, the empirical accuracy was better than theoretically expected. This indicates that one would rather underestimate than overestimate the estimation accuracy if the model is used. Small flows violate the model assumption (see (4.12) in chapter 4.6.1). This can be a reason for some deviations from the model. The right diagram in Figure 5-9 confirms that larger differences only occur for very small flows. For flows with a proportion P_f>0.002 (0.2 %), the difference stays below ±0.1.

5.5.2 Comparison of Sampling Schemes

In this section the different sampling schemes are compared. For this 1000 sampling runs are performed for each scheme and with each measurement interval. From this 1000 runs the empirical bias and the standard error for all flows in all measurement intervals are calculated for all schemes. Results are shown for a sample fraction of 5%.

5.5.2.1 Bias

The absolute and relative bias for n-out-of-N sampling was already shown in 5.5.1.1. Figure 5-10 shows the absolute and the relative empirical bias for 1-in-K sampling. There are no significant differences between the bias for 1-in-K sampling and the bias for n-out-of-N sampling.

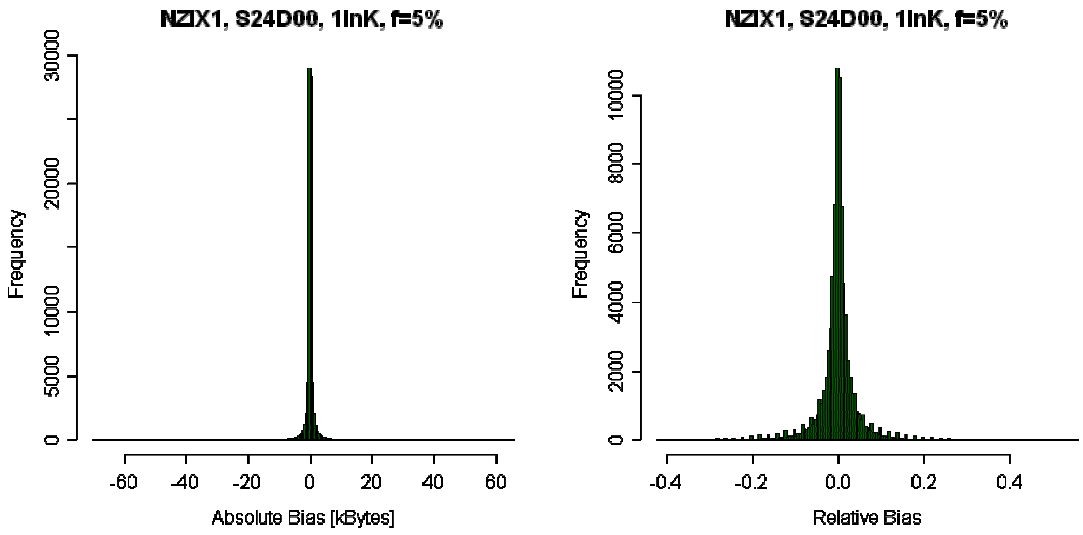


Figure 5-10: Absolute and Relative Empirical Bias for 1-in-K, f=5%

Figure 5-11 shows the empirical bias for 1-in-K sampling over the flow proportion. As for the n-out-of-N sampling the highest relative bias is observed for small flows.

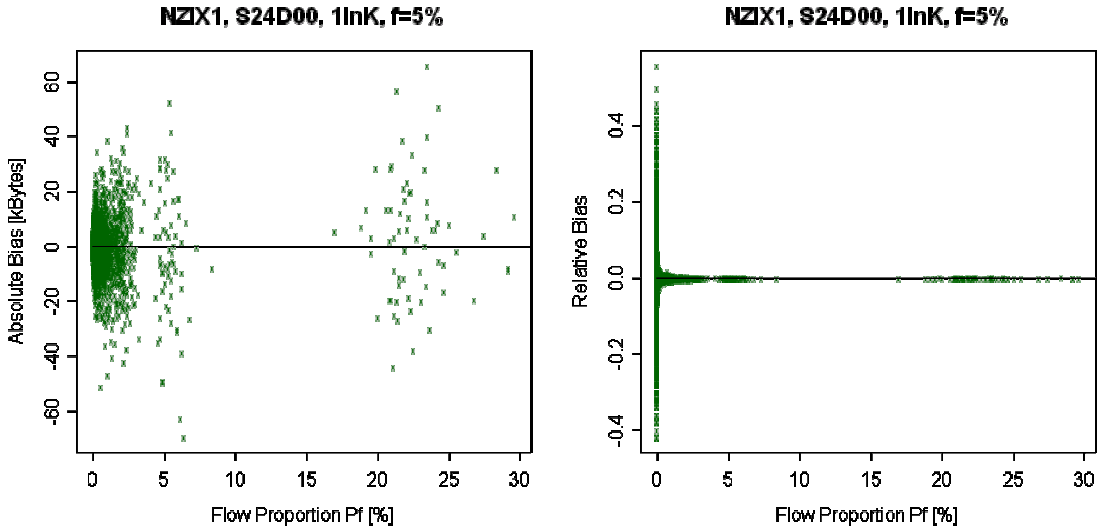


Figure 5-11: Absolute and Relative Empirical Bias for 1-in-K over Flow Proportion, f=5%

Figure 5-12 shows the absolute and the relative empirical bias for systematic sampling. Again the values are in the same range and there are no significant differences to n-out-of-N sampling.

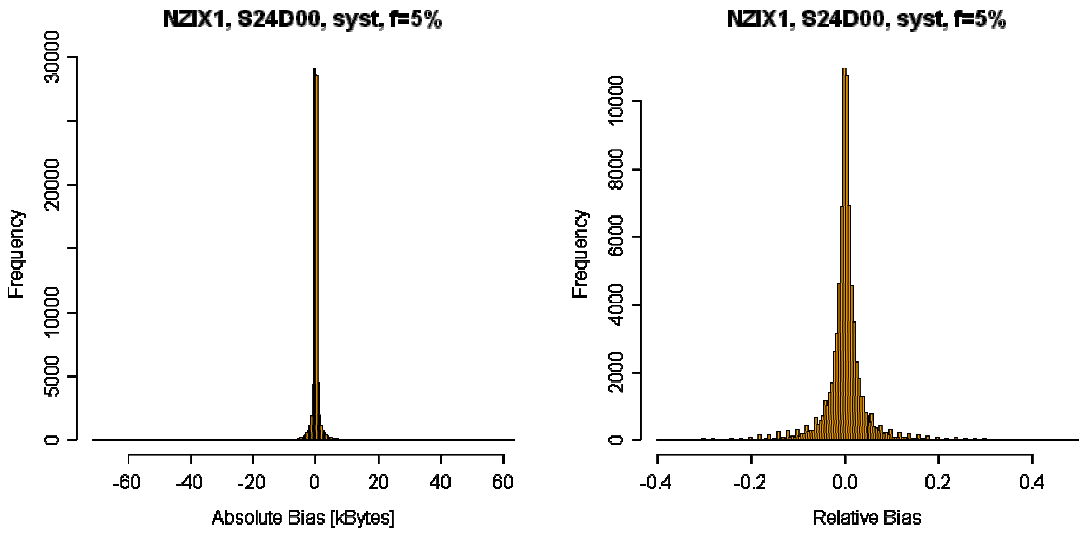


Figure 5-12: Absolute and Relative Empirical Bias for Systematic, f=5%

Figure 5-13 shows the empirical bias for systematic sampling over the flow proportion. Again the highest relative bias is observed for small flows.

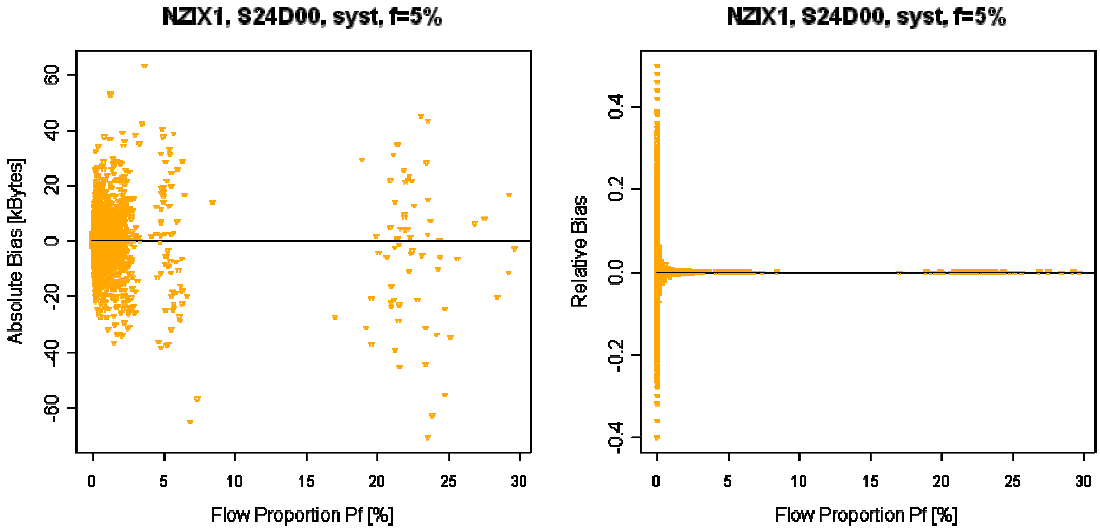


Figure 5-13: Absolute and Relative Empirical Bias for Systematic over Flow Proportion, f=5%

Table 5-4 summarizes the distributions of the bias for the investigated schemes.

Relative Bias	n-out-of-N	1-in-K	Systematic
Minimum	-0.5	-0.42	-0.40
1st Quartile	-0.01258	-0.01257	-0.01206
Median	0	-0.0000250	0
3rd Quartile	0.01239	0.01225	0.01201
Maximum	0.54	0.56	0.50
Mean	0.0000456	-0.0002312	0.0001715
StdDev	0.047	0.047	0.047

Table 5-4: Summary of Relative Empirical Bias for Different Schemes

5.5.2.2 Precision

Figure 5-14 shows the histograms of the absolute and the relative standard error for all flows in the measurement interval for n-out-of-N sampling with a sample fraction of 5%. The mean relative standard error is 1.039. The maximum is 5.33. That means for the flows with the worst accuracy the difference between estimated and real volume can be about five times the real volume. If one assumes a normal distributed estimate, the standard error provides the error boundaries for a confidence level of 68.3 % (see chapter 3.7.3). So with a probability of 68% the real volume is in the confidence limits. There is a probability of 32.7 % that the error is even larger. So the expected accuracy for some flows is definitely way to low to do reasonable accounting with a sampling fraction of 5%.

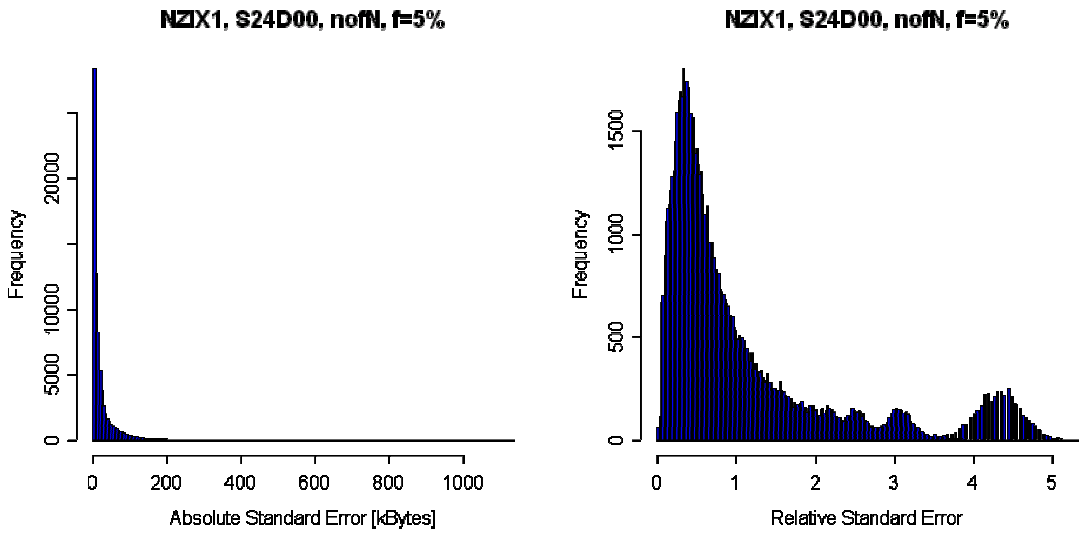


Figure 5-14: Absolute and Relative Empirical Standard Error for n-out-of-N, f=5%

Figure 5-15 shows the standard errors for n-out-of-N sampling over the flow proportion. Flows with a higher proportion (i.e. more packets) often also have a higher flow volume. Therefore the absolute standard error increases for large proportions (see section 4.6.3.3). From the second diagram it can be seen that only for the small flows the relative standard error is high. That means for small flows the accuracy is very low.

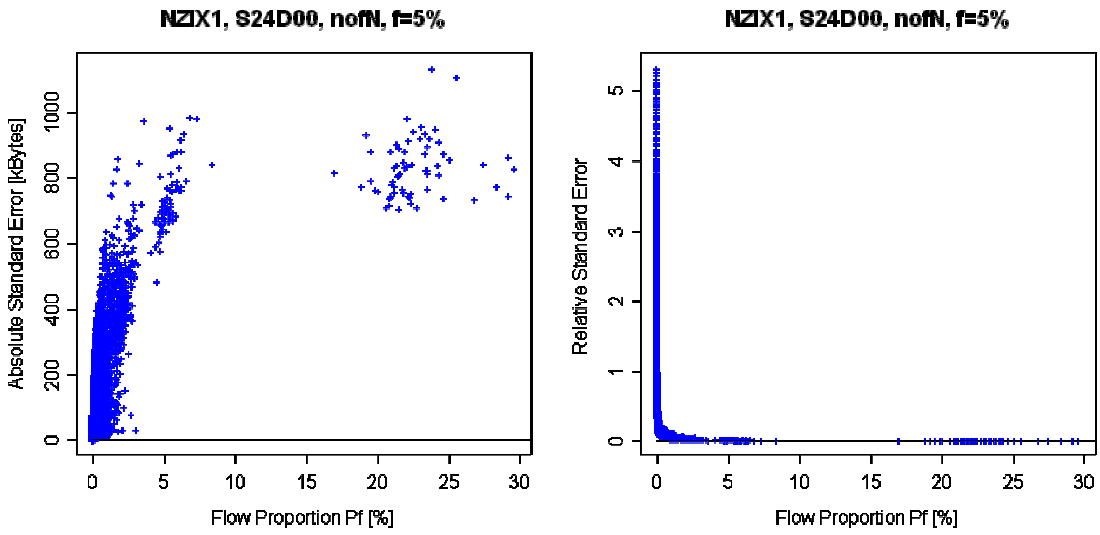


Figure 5-15: Absolute and Relative Empirical Standard Error for n-out-of-N over Flow Proportion, f=5%

The shapes of the curves for 1-in-K and systematic sampling look very similar to the n-out-of-N curves. Table 5-5 summarizes the distributions of the relative standard errors for the investigated schemes.

Relative StdError	n-out-of-N	1-in-K	Systematic
Minimum	0.01393	0.01300	0.01049
1st Quartile	0.34525	0.34371	0.33171
Median	0.62731	0.62549	0.61490
3rd Quartile	1.27593	1.27286	1.26603
Maximum	5.33183	5.36343	5.26783
Mean	1.03919	1.03617	1.02936

Table 5-5: Summary of Relative Standard Error for Different Schemes

Figure 5-16 shows the relative differences (per flow) of the empirical relative standard errors for 1-in-K (left diagram) and systematic (right diagram) sampling to the empirical relative standard errors for n-out-of-N sampling for a sample fraction of 5%. The relative difference is calculated from the empirical relative standard error from 1-in-K (or systematic) and the empirical relative standard error from n-out-of-N sampling as follows:

$$diff_{rel} = \frac{StdErr_{1inK,rel,empiric} - StdErr_{nofN,rel,empiric}}{StdErr_{nofN,rel,empiric}} \quad (5.6)$$

In contrast to formula (5.5), here the empirical results from 1-in-K are compared to empirical results from n-out-of-N and not to the theoretical expected values from the model.

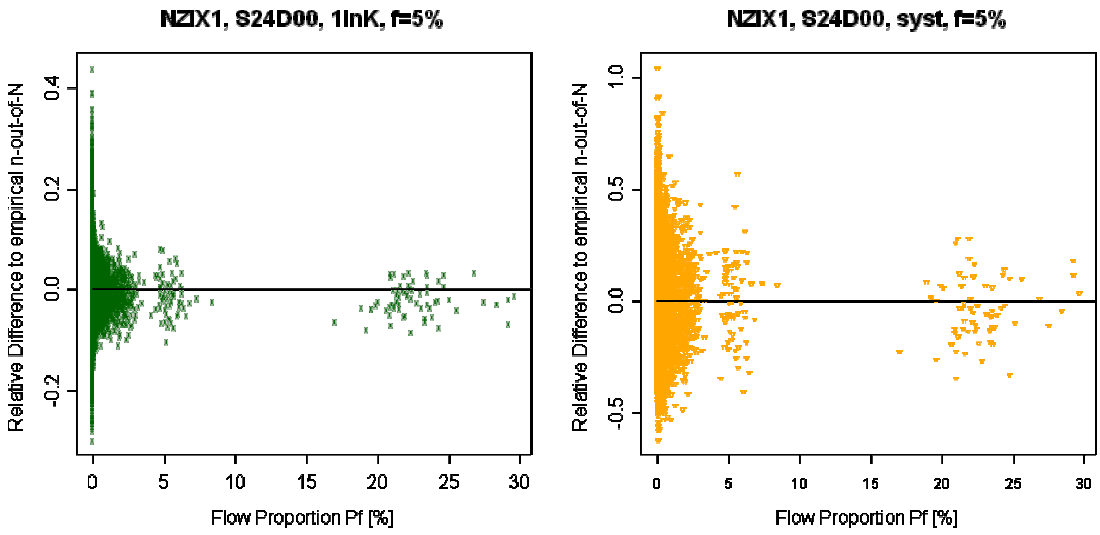


Figure 5-16: Relative Difference of 1-in-K and Systematic to Empirical Standard Error for n-out-of-N over Flow Proportion, f=5%

For systematic sampling there are positive and negative differences for all flow sizes. For 1-in-K sampling more negative differences can be observed, especially for large flows. For both schemes the differences get smaller for large flows, i.e. the accuracy approaches the accuracy achieved with n-out-of-N sampling. Table 5-6 summarizes the distributions of the differences.

Relative Difference	1-in-K	Systematic
Minimum	-0.2988	-0.62240
1st Quartile	-0.0308	-0.1102
Median	-0.0031	-0.0238
3rd Quartile	0.0249	0.0763
Maximum	0.4401	1.0410
Mean	-0.0020	-0.0118
StdDev	0.0488	0.1498

Table 5-6: Summary of Relative Difference to Empirical n-out-of-N

For both schemes the majority of flows get a negative difference, i.e. a higher accuracy than for n-out-of-N sampling. 42,225 (53.19 %) of all 79,383 flows in the trace get a better accuracy with 1-in-K sampling than with n-out-of-N sampling. 45,215 flows (56.96 %) get a better accuracy with systematic sampling than with n-out-of-N sampling.

The differences for 1-in-K sampling are smaller than then the differences for systematic sampling. That means that the accuracy for 1-in-K is more close to the accuracy of n-out-of-N.

5.5.3 Influence of Sample Fraction

In this section the influence of the sample fraction on the estimation accuracy is investigated for the investigated sampling schemes (n-out-of-N, 1-in-K, systematic). Since the standard error differs for each flow, three flows with very different characteristics are selected exemplarily for an in-depth investigation. The flows are selected from the first measurement interval. Flow 15 contains only 71 packets and therefore is a comparatively small flow. Flow 6 is a medium size flow with 3,960 packets, but has a quite small packet size standard deviation. Flow 29 is a large flow with 25,426 packets and a large packet size mean and packet size standard deviation. Table 5-7 shows the characteristics of the selected flows.

Flow Number	# Packets	Volume [Bytes]	Mean Packet Size [Bytes]	Packet Size Standard Deviation [Bytes]
15 (small)	71	4,596	64.73	103.30
6 (medium)	3,960	271,429	68.54	13.58
29 (large)	25,426	12,772,042	502.32	582.77

Table 5-7: Selected Flows

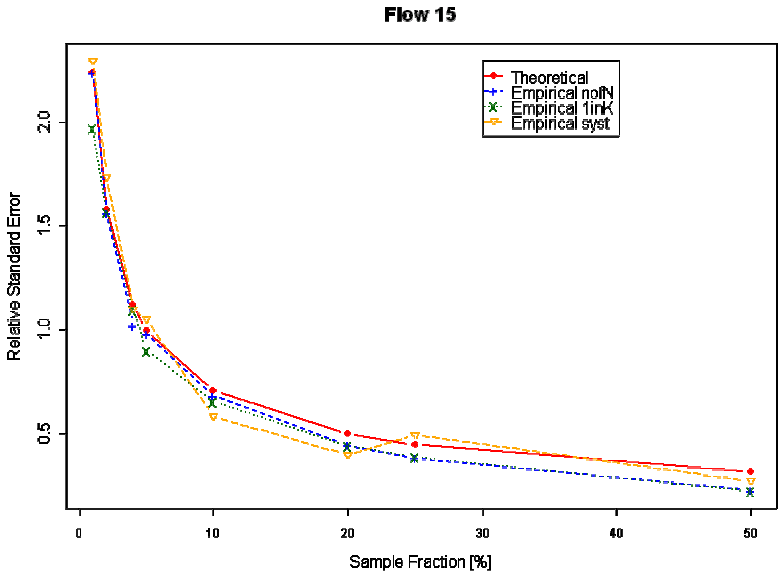


Figure 5-17: Comparison of Schemes for Different Sample Fraction (Small Flow)

Figure 5-17 shows the relative standard error for the small flow 15. It compares the theoretical values, calculated from the model (red solid line) with the empirical values for n-out-of-N (blue dashed line), 1-in-K (green dotted line) and systematic sampling (yellow long-dashed line). The standard error is pretty high. For small sample fractions the relative standard error gets larger than 2. That means the estimate can be twice the real value, even if only a confidence level of 68.3% is assumed. This is way too high for using these results for usage-based accounting.

Although the flow does not fulfill the initial model assumptions, the empirical values for n-out-of-N sampling are very close to the theoretical values from the model for small sampling fractions. As expected the accuracy increases (standard error decreases) if a higher sample fraction is used. For larger sampling fractions the empirical values are smaller than the theoretically expected value. This is because one of the initial assumptions that had to be made to derive the model was that the sampling fraction is below 5 %. The empirical standard error for 1-in-K sampling for most sample fractions is slightly smaller than the theoretical or empirical standard error for n-out-of-N sampling. This means for those sampling fractions a higher accuracy is achieved with 1-in-K sampling. For large sample fractions the standard error of 1-in-K sampling gets close to the n-out-of-N results. The systematic sampling performs sometimes better and sometimes worse than the other schemes. For three sample fractions the accuracy for systematic sampling is worse than expected by the model, whereas n-out-of-N and 1-in-K always perform equally well or better than the model.

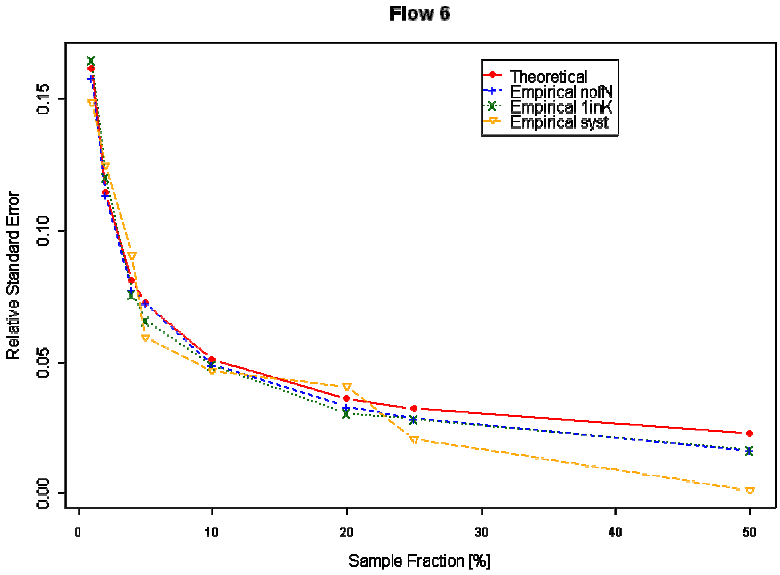


Figure 5-18: Comparison of Schemes for Different Sample Fraction (Medium Flow)

Figure 5-18 shows the comparison of the results for the investigated schemes for the medium size flow 6. The standard error is much smaller for this flow. This conforms to the theoretical considerations that larger flows get a better accuracy. Again the n-out-of-N and 1-in-K results are very close with a bit better results for 1-in-K for most sample fractions. Systematic sampling performs sometimes better and sometimes worse.

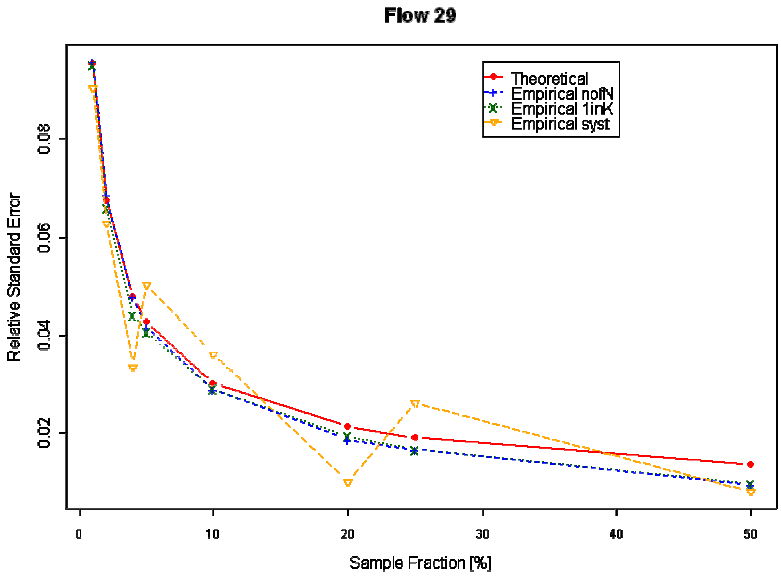


Figure 5-19: Comparison of Schemes for Different Sample Fraction (Large Flow)

Figure 5-19 shows the comparison of sampling methods for the large flow 29. Although the packet sizes in the flow have a quite high variance (see Table 5-7), the standard error is much lower than for the smaller flows. So the flow size (number of packets per flow) here has the major influence on the accuracy. For the systematic sampling there are quite large positive and negative differences to the model and to the other schemes.

5.5.4 Approximation of Standard Error from Samples

The real flow characteristics are not known when sampling is deployed. Therefore one has to approximate the standard error from the traffic parameters estimated from the samples. In these experiments the traffic parameters of each flow were estimated from the samples. The traffic parameters of interest are the parameters that are needed as input to the formula: the number of packets in the flow, the mean packet size and the packet size variance. For each run different packets are sampled, therefore one gets different estimations for the traffic parameters per run. The estimation of the traffic parameters is more accurate the more samples are taken. That means one expects to get a more precise approximation of the standard error if the sample fraction is large.

The following figures show the theoretical and empirical standard error for n-out-of-N and 1-in-K sampling compared to the range of approximated standard errors. All approximated standard errors are calculated with the n-out-of-N model and estimated flow characteristics. For this the input parameters of the model are estimated by using the sampled values from the run. Since for each run different packets are selected, one gets different estimates of the traffic parameters for each run and therefore also a different approximated standard error per run. The distributions of the standard errors for different runs for one sample fraction are shown as boxplots in the diagrams below.

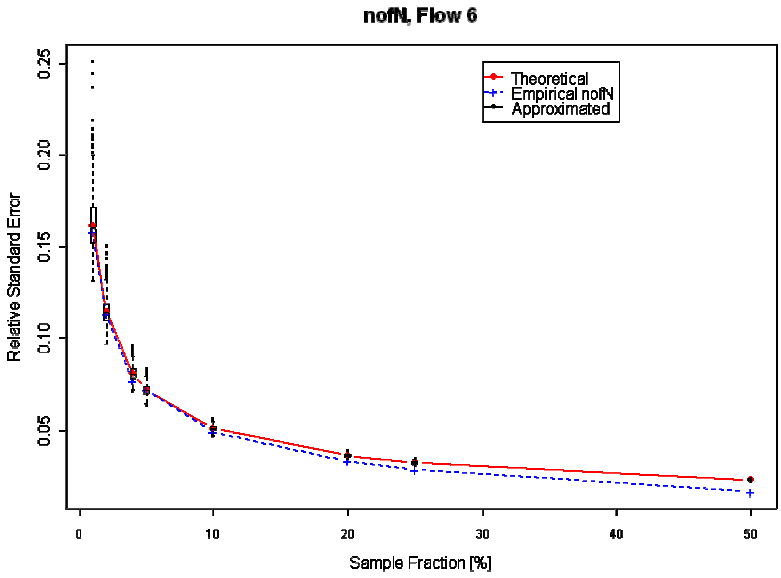


Figure 5-20: Approximated Standard Errors (Medium Flow)

Figure 5-20 shows the theoretical expected (red solid line) and the empirical standard errors (blue dashed line) for the medium flow 6 for n-out-of-N sampling and the boxplots for the approximated standard error from the 1000 sampling runs. From the boxplots one can see how the approximated standard errors for the different runs vary. For small sampling fractions the approximated standard errors varies much more than for large sample fractions. That means that one gets an inaccurate assumption about the accuracy of the flow if the sample fraction is small and the standard error as to be approximated from few values. Nevertheless for the medium size flow most approximated values are close to the achieved accuracy.

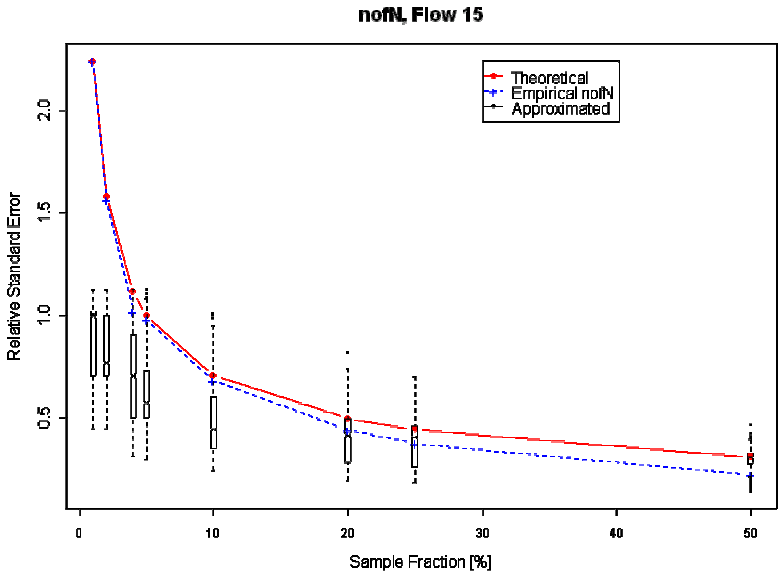


Figure 5-21: Approximated Standard Errors Sample Fraction (Small Flow)

The diagram for the very small flow 15 shows a quite different picture (Figure 5-21). The approximation of the standard error is much worse than for the medium size flow, especially

for small sampling fractions. In most cases the standard error is underestimated, i.e. one would assume a too high accuracy for this flow if the accuracy is approximated from sampled values. The reason for this is that for small sample fractions often none or only few packets of the flow are part of the sample. The few packets are often insufficient to provide a reasonable estimate for mean packet size and packet size variance.

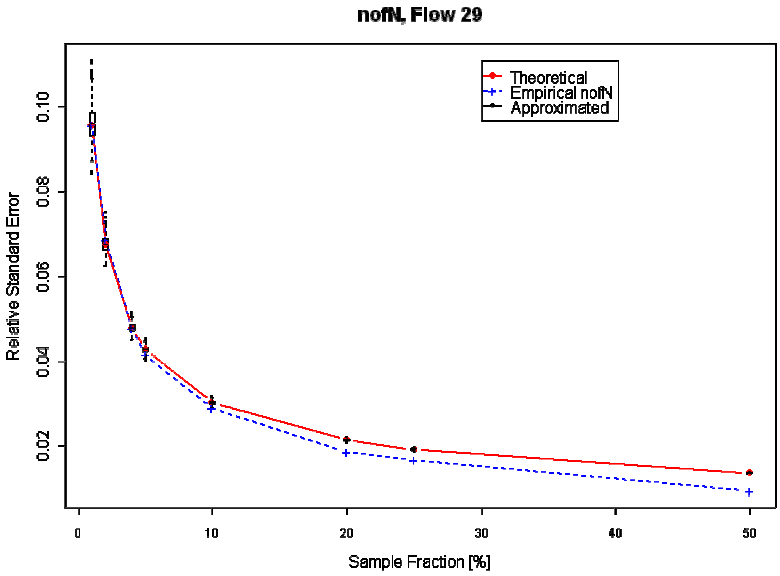


Figure 5-22: Approximated Standard Errors (Large Flow)

The results for the large flow 29 are much better (Figure 5-22). Already for small sample fractions one can provide a good approximation for the accuracy when estimating input parameters for the model from the sampled packets.

5.6 Comparison of Results with Accuracy Requirements

Table 5-8 shows the maximum relative standard error for different accuracy requirements for a normal distributed estimate. Please note that the estimation error can be exceeded. The accuracy requirement only says that the in 95% or 99% of the cases the error will not exceed the given value.

Rel. Estimation Error	CI Level	Max rel. StdErr
0.01 (1%)	99% (z_c=2.58)	0.003876
0.01 (1%)	95% (z_c=1.96)	0.005102
0.05 (5%)	99% (z_c=2.58)	0.019380
0.05 (5%)	95% (z_c=1.96)	0.025510
0.1 (10%)	95% (z_c=1.96)	0.051020
0.15 (15%)	95% (z_c=1.96)	0.076531
0.20 (20%)	95% (z_c=1.96)	0.102041
0.30 (30%)	95% (z_c=1.96)	0.1531

Table 5-8: Maximum relative Standard Error for Different Accuracy Requirements

One can see that the relative standard error has to be below 0.05 in order to achieve a relative estimation error of 0.1 with a confidence level of 95%.

Figure 5-23 shows the conformance of the flows to the accuracy requirements for the fine grained classification S24D24 and n-out-of-N sampling. One can see that the number of packets is the extremely relevant for the accuracy. A high accuracy (green triangles) can only be achieved for large flows. The reason for this is that the proportion of packets from a specific flow in the measurement interval defines the probability that the sample contains packets of this flow. Furthermore, higher accuracies can be achieved for flows with small packet size variance.

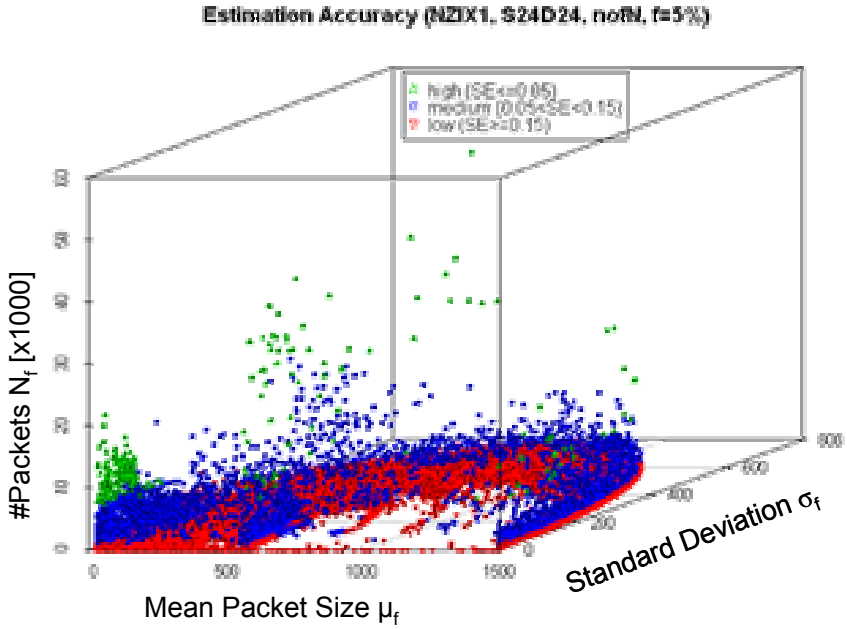


Figure 5-23: Conformance to Accuracy Requirements (S24D24, n-out-of-N, f=5%)

Figure 5-24 shows the conformance of the flows to the accuracy requirements for the more coarse grained classification S24D00. Here the relevance of the number of packets per flow becomes even clearer. For all the large flows a high accuracy is achieved whereas the small flows only get a very small accuracy.

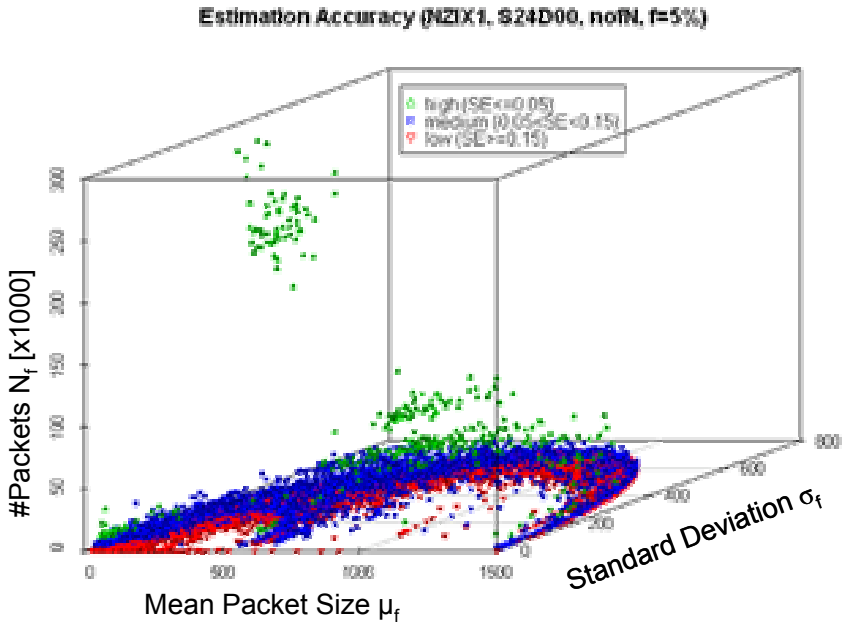


Figure 5-24: Conformance to Accuracy Requirements (S24D00, n-out-of-N, f=5%)

Table 5-9 show how many flows in the trace conform to given accuracy requirements for the S24D00 classification and different schemes. Accuracy requirements are given as an estimation error margin and a confidence level. The results are specific for the given flow definition based on source network and the chosen measurement interval length. The first column shows the maximum relative standard error. The second column shows the

corresponding accuracy described by the relative estimation error and the confidence level (CL) for a normal distributed estimate. The values in the table denote the number of flows for which the empirical relative standard error is below the given standard error in that row and larger than the standard error in the previous row.

Max rel. StdErr	Error/CL	nofN	1inK	Syst
0.003876	0.01/99%	0	0	0
0.005102	0.01/95%	0	0	0
0.019380	0.05/99%	64	64	62
0.025510	0.05/95%	8	8	21
0.051020	0.1/95%	401	403	484
0.076531	0.15/95%	933	950	1013
0.102041	0.2/95%	1124	1143	1280
0.1531	0.3/95%	2830	2829	2939
> 0.1531	-	74023	73986	73584

Table 5-9: Conformant Flows (S24D00, f=5%)

For the vast majority of flows the estimation accuracy is way too low to use it as basis for accounting. This is the case for all three schemes. For 1-in-K and for systematic sampling a few more flows achieve higher accuracy requirements than for n-out-of-N. But also for those schemes only a small fraction of flows achieve a reasonable accuracy.

In order to get a higher accuracy per flow one can increase the sample fraction, work with more coarse grained classification or modify the measurement interval length. When modifying the measurement interval length it is relevant how flow characteristics evolve in order to assess the accuracy (as shown in section 4.6.3.2). Approaches for this are currently investigated in the project VEGAS-II.

5.7 Conclusion

In this chapter the trace analysis and the sampling experiments with the NZIX trace were described. The analysis of the complete trace showed how the traffic characteristics vary for the different flows.

The trace was investigated with two different classifications, one with source and destination network and one with the source network only. With both classification schemes only a few large flows were observed in the trace. The majority of flows are small. This matches observations made by other researchers (see references in 2.1.2). The different flow characteristics lead to different theoretical accuracies for individual flows. Empirical investigations with 1000 runs with n-out-of-N sampling showed that empirical results are close to the values expected by the model.

One initial assumption for deriving the model was that a flow should consist of at least 10% of the overall traffic. With the selected classification only very few flows in the whole trace conform to this assumption. Nevertheless, the experiments showed that empirical results are close to the model already for flows with a proportion of 0.2 % of the overall traffic.

For much smaller flows (<0.2%) the model cannot be applied. It is questionable whether providers want to account for such small flows separately. For such cases it makes much more sense to work with a more coarse classification to get larger flows or to collect all flows that are too small in one class. With the last approach, providers can assess the revenue loss caused by neglecting the transmitted volume of the small flows. One also could apply a combination of packet sampling with flow sampling approaches shown in 3.6.1, to neglect flows that are too small for achieving a sufficient accuracy. Approaches for this are currently investigated in the project VEGAS-II.

Another assumption was that the sample fraction has to be below or equal 5% to apply the model. And indeed, empirical values differ from the model for higher sample fractions. In the experiments it was shown that the standard error gets smaller than expected by the model for sample fractions above 5%. Therefore the achievable accuracy is underestimated if the model is used for higher sample fraction. Assuming a too low accuracy is usually less critical than an overestimation of the accuracy.

The results for 1-in-K sampling were in most cases close to the n-out-of-N results. An In-depth investigations of individual flows showed that this is true also for different sample fractions. A direct comparison of both schemes at a sample fraction of 5% showed that especially for large flows often a slightly higher accuracy can be achieved with 1-in-K sampling. That means that some accuracy gain is achieved by stratification in accordance to the packet count. According to the experimental results one would rather underestimate the accuracy when using the n-out-of-N model for the assessment of 1-in-K sampling.

The systematic sampling also performed better in terms of the number of flows that get a better accuracy. Nevertheless, the precision values vary very much and are quite unpredictable. When looking at individual flows the accuracy differed much to the theoretical and empirical results for 1-in-K and n-out-of-N sampling. Since potential correlations can never be excluded, results for systematic sampling are trace-specific and cannot be generalized for arbitrary traces.

A comparison to common accuracy requirements showed that only for very large flows a reasonable accuracy can be achieved. The main influencing parameter is the proportion of packets from the specific flow to all packets in the measurement interval. A more coarse grained classification or a modification of the measurement interval length can help to achieve higher accuracies.

The approximation of the standard error from the sampled data is necessary in reality, because the flow characteristics are unknown. For this the traffic characteristics are estimated from the

sampled packets and those estimates are used as input parameters for the model. In the experiments it could be observed that the approximation can vary extremely for very small sample fractions and small flows but approaches the empirical values for larger flows and if sample sizes are larger.

6 Sampling Techniques for SLA Validation

Service Level Agreements (SLAs) are contracts between customers and providers that define the quality of the transport of packets through the provider network. As motivated in 2.3.1 passive measurements are well suited to validate the fulfillment of SLA guarantees.

6.1 Assessment and Selection of Schemes

The customer traffic observed at the entry and exit point of the provider network forms the population. It is assumed that the sampling process works on the traffic mix of the customer who contracted the SLA. As explained above one-way delay is used as example metric due to its importance for many applications.

The main difference between the SLA validation and the accounting scenarios is that some metrics for SLA validation may require multipoint measurements. That means information about each selected packet need to be transmitted to the point were the metric calculation takes place. That means additional resources for data transfer are required. Furthermore, if sampling takes place at both points, the selection of packets should be synchronized (see 3.5). The following schemes are selected for an in-depth investigation for SLA validation.

C/C/S: Systematic count-based sampling can be implemented and operated with minimal effort. Due to packet loss, delay and re-ordering the sequence of packets at different observation points differ. Therefore this scheme is only applicable if sampling is applied only at selected observation points (see heterogeneous measurement approach in 3.5). The scheme is investigated in order to compare whether with this simple method a similar accuracy can be achieved as with a more complex random selection.

C/C/RN: n-out-of-N sampling is a random sampling scheme that can easily be implemented and requires only few resources. As for systematic sampling a synchronization of selection processes between multiple observation points cannot be achieved due to potential packet loss and reordering. Therefore the scheme can be also only applied in heterogeneous scenarios, were only dedicated measurement points apply sampling.

C/-/RP: Uniform probabilistic sampling is of interest for multipoint measurements. Whereas it is in general not possible to synchronize two n-out-of-N sampling processes at different observation points, probabilistic sampling can be emulated by a hash-based selection, which allows a multipoint synchronization [DuGr00]. But due to variations of the sample size it is likely that probabilistic sampling performs worse than n-out-of-N sampling. Therefore it here is investigated what accuracy can be achieved with uniform probabilistic sampling compared to n-out-of-N sampling.

In addition to the schemes above the following stratified scheme is investigated:

Content-based stratified C/C/RN sampling stratifies the population in accordance to packet attributes derived from the content. The calculation of some metrics (e.g., one-way-delay) requires the transfer of per-packet information. Therefore the effort for calculating this metric

for each sampled packet is very high. If one can find another packet attribute that can be investigated with less effort (e.g., from header fields) that has some correlation with the investigated metric (here delay) one can achieve a stratification gain and achieve the same estimation accuracy with less effort or a higher accuracy without additional effort.

The schemes below are ***not*** considered for an in-depth investigation due to the following reasons:

T/Co/RP, C/Co/RP: Non-uniform probabilistic sampling based on packet content could be useful to realize a biased selection. But the sample size for probabilistic sampling is variable (see 4.3.1). With stratified sampling it is possible to realize a biased selection with a controllable sample size. Therefore this work concentrates on stratified schemes and content-based non-uniform probabilistic schemes are not considered.

T/Co/S, C/Co/S: Filtering is a deterministic content-based selection. In contrast to packet sequence or arrival time, most of the packet content is the same at all observation points. Therefore it is possible to synchronize selection processes at different observation point by triggering the selection based on the packet content (see 3.5). Hash-based selection methods are a special form of filtering that allows the emulation of a random probabilistic selection. The quality of an emulation of probabilistic sampling with hash-based selection methods is still subject to research ([DuGr00], [NiMD04], [NiMT04]). First recommendations for schemes are collected by the PSAMP group in [ZsMD05]. Nevertheless, achieving a sufficient independence of the selection with regard to different packet attributes is a problem because hash-based methods are a deterministic content-based selection and therefore extremely receptive to bias. The emulation of random sampling with hash functions is an interesting related topic, which is addressed by several recent publications. It is expected that soon results from others can be exploited. Therefore hash-based selection methods (T/Co/S, C/Co/S) themselves are not considered in this work. Nevertheless it is investigated whether there is a differences between probabilistic sampling, which can be emulated by a hash-based selection, and n-out-of-N sampling (see C/-/RP).

For the remaining schemes that are not considered the same reasons apply as stated in 4.3.

6.2 Proportion vs. Percentile Estimation

The goal of SLA validation is to validate if the packets in a flow are conformant to QoS guarantees (e.g., delay, jitter) given in an SLA. An estimation of the whole distribution of the metric of interest is difficult and contains much more information than needed. The estimation of ***mean*** and ***standard deviation*** gives first insights about the quality situation, but is inadequate to validate the SLA conformance.

The ***percentiles*** of the distribution reveal the value below which one can assume the majority (e.g., 95%) of observed values [ChMC03]. It provides a valuable parameter for assessing the general network situation but is unsuitable to quantify non-conformance. If the percentile lies

above the defined threshold, the approach does not provide information about the percentage of packets that really violated the contract.

Therefore I here propose a different approach. Instead of estimating percentiles, I propose to estimate the percentage of non-conforming packets. With this one can formulate the SLA validation as estimation of the ***proportion*** of packets that exceed a predefined threshold t. Packets with a value equal or above the threshold are considered as violators (hit,1), packets with values below the threshold are considered conformant (no-hit, 0). The number of non-conformant packets (violators) obtained after that classification can be modeled as Binomial distributed random variable.

As mentioned above, one-way delay is used as example metric. Nevertheless the techniques and mathematical models introduced here can also be used to estimate the proportion of packets with other attributes (e.g., proportion of large packets, proportion of packets from a specific flow, etc.).

6.3 Proportion Estimation with n-out-of-N Sampling

6.3.1.1 Calculating an Estimate

The metric of interest is the proportion of packets with a specific property (here "violate SLA"). The real proportion is given by the number M of violators in the measurement interval divided by the number N of all packets in the measurement interval:

$$P = \frac{M}{N} \tag{6.1}$$

The proportion of violators in the sample is used as estimate for the proportion of violators in the population (method of moments). For this, the number m of violators in sample is divided by the real sample size n_R.

$$\hat{P} = \frac{m}{n_R} \tag{6.2}$$

Let the random variable x_i denote whether the i^{th} sampled packets conforms to the SLA or not (e.g., $x_i = 0$ if the packet delay $d < d_{max}$ and $x_i = 1$ if $d \geq d_{max}$). Then x_i has only two possible outcomes and follows a Bernoulli distribution with probability of success $P=M/N$. With this the number m of violators in the sample can be modeled as number of hits in an experiment with n_R trials.

$$m = \sum_{i=1}^{n_R} x_i \qquad \text{with} \qquad x_i \sim Be(P) \tag{6.3}$$

With n-out-of-N sampling one selects exactly n packets out of the population of the N packets observed in the measurement interval. That means the real sample size n_R remains constant and equals the target sample size n_T.

$$n_R = n_T = const. \tag{6.4}$$

Since one cannot select a packet again that already was selected, one has to consider a selection without replacement, i.e., *m* has to be considered as random variable with a hyper geometric distribution. Therefore expectation and variance of *m* can be expressed as follows:

$$E[m] = n_T \cdot P \tag{6.5}$$

$$V[m] = \frac{N - n_T}{N - 1} \cdot n_T \cdot P \cdot (1 - P) \tag{6.6}$$

6.3.1.2 Estimation Bias and Accuracy

With (6.5) and (6.6) and since n_T is constant for n-out-of-N sampling one gets the following expectation and variance for the estimate $\hat{P}$:

$$E\left[\hat{P}\right] = E\left[\frac{m}{n_T}\right] = \frac{1}{n_T} \cdot E[m] = \frac{1}{n_T} \cdot n_T \cdot P = P \tag{6.7}$$

$$V\left[\hat{P}\right] = V\left[\frac{m}{n_T}\right] = \frac{1}{n_T^2} \cdot V[m] = \frac{1}{n_T^2} \cdot \frac{N - n_T}{N - 1} \cdot n_T \cdot P \cdot (1 - P) = \frac{1}{n_T} \cdot \frac{N - n_T}{N - 1} \cdot P \cdot (1 - P) \tag{6.8}$$

Since the expectation equals the real proportion one gets an unbiased estimate. With $\frac{N - n_T}{N - 1} \approx \frac{N - n_T}{N} = 1 - \frac{n_T}{N} = 1 - f_T$ the absolute and relative standard error can be derived as follows:

$$StdErr_{abs} = \sqrt{\frac{P \cdot (1 - P)}{n_T}} \cdot \sqrt{1 - \frac{n_T}{N}} = \sqrt{\frac{P \cdot (1 - P) \cdot (1 - f_T)}{n_T}} \tag{6.9}$$

$$StdErr_{rel} = \frac{StdErr_{abs}}{P} = \frac{1}{P} \cdot \sqrt{\frac{P \cdot (1 - P) \cdot (1 - f_T)}{n_T}} = \sqrt{\frac{(1 - P) \cdot (1 - f_T)}{P \cdot n_T}} \tag{6.10}$$

The estimation accuracy depends on the sample fraction and on the real violator proportion. The real violator proportion is unknown and has to be approximated from the sample or replaced by worst case parameters in order to make an accuracy prediction in advance (see section 6.8). If the sample fraction is small ($f_T < 5\%$), one can neglect the influence of the selection without replacement, approximate the hyper geometric random variable by a binomial distributed random variable and therefore neglect the finite population correction factor. With this one gets the following simplified formulas:

$$StdErr_{abs} = \sqrt{\frac{P \cdot (1 - P)}{n_T}} \tag{6.11}$$

$$StdErr_{rel} = \sqrt{\frac{(1 - P)}{P \cdot n_T}} \tag{6.12}$$

6.4 Proportion Estimation with Probabilistic Sampling

With probabilistic sampling each packet is selected with a given probability regardless of the fact how many packets have been already selected before. Therefore the real sample size n_R

varies for each run, and in most cases will not be equal to the target sample size n_T. It is expected that this effect gets smaller for longer measurement intervals because n_R approaches n_T for large populations. One here has two possibilities for calculating an estimate, depending on the amount of knowledge that is available. Therefore the following two cases are distinguished:

- Case A: Extrapolation with target sample size: If one doesn't know how many packets were sampled one has to extrapolate with the target sample size n_T.
- Case B: Extrapolation with real sample size: If one can gain knowledge about the exact sample size n_R (e.g., by providing an additional packet counter), the exact sample size can be used for calculating the estimates.

6.4.1 Case A: Extrapolation with Target Sample Size

Case A addresses the extrapolation with the target sample size.

6.4.1.1 Calculating an Estimate

If one uses the target sample size for extrapolation one gets the following estimate:

$$\hat{P} = \frac{m}{n_T} \tag{6.13}$$

Nevertheless, the number m of violators that occur in the sample can be still calculated as follows.

$$m = \sum_{i=1}^{n_R} x_i \tag{6.14}$$

The difference to n-out-of-N sampling is that n_R can vary for each sample run. It can differ from the target sample size n_T and cannot be considered as constant. Since n_R can differ from n_T a higher estimation error is expected than with the extrapolation with the real sample fraction.

6.4.1.2 Estimation Bias and Accuracy

An approach for modeling probabilistic packet sampling by neglecting the variability of the real sample fraction is shown in [DuLT02]. In the paper the model is used to estimate the amount of packets that belong to a specific flow. In this approach the variability of the sample size is neglected and the calculation is done based on the target sample size n_T.

The number of packets that belong to a specific flow is estimated by modeling the selection process with a Bernoulli distributed random variable ω_i with success probability $f_T=n_T/N$. ω_i becomes 1 if the packet is selected and 0 if the packet is not selected.

If one considers the packet property "violate SLA" instead of "belong to flow f", one can apply the same model and can express the number of violators in the sample as follows[17]:

$$m = \sum_{i=0}^{M} \omega_i \quad \text{with} \quad \omega_i \sim Be(f_T) \tag{6.15}$$

Since ω_i is Bernoulli distributed it has the following expectation and variance:

$$E[\omega_i] = f_T \tag{6.16}$$

$$V[\omega_i] = f_T \cdot (1 - f_T) \tag{6.17}$$

With this one can calculate the expectation and variance of the number m of violators in the sample.

$$E[m] = E\left[\sum_{i=0}^{M} \omega_i\right] = \sum_{i=0}^{M} E[\omega_i] = M \cdot E[\omega_i] = M \cdot f_T = M \cdot \frac{n_T}{N} \tag{6.18}$$

$$V[m] = V\left[\sum_{i=0}^{M} \omega_i\right] = \sum_{i=0}^{M} V[\omega_i] = M \cdot V[\omega_i] = M \cdot f_T \cdot (1 - f_T) \tag{6.19}$$

From this the expectation and variance of the estimate can be derived as follows:

$$E\left[\hat{P}\right] = E\left[\frac{m}{n_T}\right] = \frac{1}{n_T} \cdot E[m] = \frac{1}{n_T} \cdot M \cdot \frac{n_T}{N} = \frac{M}{N} = P \tag{6.20}$$

$$V\left[\hat{P}\right] = V\left[\frac{m}{n_T}\right] = \frac{1}{n_T^2} \cdot V[m] = \frac{1}{n_T^2} \cdot M \cdot f_T \cdot (1 - f_T)$$

$$= \frac{1}{n_T^2} \cdot M \cdot \frac{n_T}{N} \cdot (1 - f_T) = \frac{P \cdot (1 - f_T)}{n_T} \tag{6.21}$$

So by neglecting the variability of the sample size, one theoretically gets an unbiased estimate with the following absolute and relative standard error.

$$StdErr_{abs} = \sqrt{\frac{P \cdot (1 - f_T)}{n_T}} \tag{6.22}$$

$$StdErr_{rel} = \frac{StdErr_{abs}}{P} = \frac{1}{P} \cdot \sqrt{\frac{P \cdot (1 - f_T)}{n_T}} = \sqrt{\frac{(1 - f_T)}{P \cdot n_T}} \tag{6.23}$$

6.4.2 Case B: Extrapolation with Real Sample Size

If one is able to work with the real sample size n_R more accurate estimates are expected. But predictability of the precision becomes difficult because n_R is only known after the sampling process and therefore has to be considered as random variable when calculating the estimation accuracy.

[17] A different notation is used than in [DuLT02] to be consistent with the notation throughout this document.

6.4.2.1 Calculating an Estimate

If one has information about the real sample size, e.g., because an additional packet counter is provided at the observation point, one can calculate the estimate of the violator proportion as for the n-out-of-N sampling:

$$\hat{P} = \frac{m}{n_R} \tag{6.24}$$

When the sampling process has completed for a specific measurement interval, the estimate can easily be calculated by analyzing how many packets were selected (n_R) and how many of the selected packets violated the SLA.

6.4.2.2 Bias and Precision of the Estimate

Like in 6.4.1.2, one has to look at the expectation and the variance of the estimate $\hat{P}$, in order to asses the estimation quality:

$$E\left[\hat{P}\right] = E\left[\frac{1}{n_R} \cdot \sum_{i=1}^{n_R} x_i\right] \tag{6.25}$$

$$V\left[\hat{P}\right] = V\left[\frac{1}{n_R} \cdot \sum_{i=1}^{n_R} x_i\right] \tag{6.26}$$

It was shown in 6.4.1.2 how expectation and variance of m can be calculated. Nevertheless, for an extrapolation with the real sample fraction it has to take into account that the sample size n_R is a random variable itself. So for the calculation of the expectation and variance of the estimate $\hat{P}$ one has to divide the random variable m by the random variable n_R.

$$E\left[\hat{P}\right] = E\left[\frac{m}{n_R}\right] \tag{6.27}$$

$$V\left[\hat{P}\right] = V\left[\frac{m}{n_R}\right] \tag{6.28}$$

This leads to a problem. The number of selected violators m heavily depends on the number of selected packets n_R in the measurement interval. So the problem is to calculate the expectation and variance of a division of two dependent random variables. Since the correlation between m and n_R is unknown, the variance cannot be calculated for this case.

6.4.2.3 Variability of Real Sample Size for Probabilistic Sampling

According to theory the distribution of n_R should approach a binomial distribution, because it can be modeled as number of hits in N Bernoulli experiments with success probability n_T/N. The expected mean and variance are:

$$E\left[n_R\right] = N \cdot \frac{n_T}{N} = n_T \tag{6.29}$$

$$V\left[n_R\right] = N \cdot \frac{n_T}{N} \cdot \left(1 - \frac{n_T}{N}\right) = n_T \cdot \left(1 - \frac{n_T}{N}\right) \tag{6.30}$$

6.5 Proportion Estimation with Systematic Sampling

If all packet delays in the measurement interval were independent, systematic sampling would equal random sampling. For systematic count-based sampling one could apply the same mathematical model results as for n-out-of-N sampling.

If correlations occur, the systematic selection process can interfere with periodicities in the packet sequence. In such cases one may get a non-representative accumulation of packets with specific properties (e.g., packets with high delays) in the sample and with this a biased estimation. The nature of this bias heavily depends on the specific traffic mix. Therefore one cannot derive a generic model (valid for arbitrary traces) for the accuracy as done for the random selection methods.

6.6 Theoretical Comparison of Schemes

Table 6-1 shows the absolute and relative standard errors for the different sampling schemes as derived in section 6.3, and 6.4.

Sampling Method	**Conditions**	**Absolute Standard Error**	**Relative Standard Error**
n-out-of-N (approximated)	$f_T \leq 5\%$, finite population correction neglected	$StdErr_{abs} = \sqrt{\frac{P \cdot (1-P)}{n_T}}$	$StdErr_{rel} = \sqrt{\frac{(1-P)}{P \cdot n_T}}$
n-out-of-N		$StdErr_{abs} = \sqrt{\frac{P \cdot (1-P) \cdot (1-f_T)}{n_T}}$	$StdErr_{rel} = \sqrt{\frac{(1-P) \cdot (1-f_T)}{P \cdot n_T}}$
Probabilistic Case A [DuLT02]	Variability of n_R neglected	$StdErr_{abs} = \sqrt{\frac{P \cdot (1-f_T)}{n_T}}$	$StdErr_{rel} = \sqrt{\frac{(1-f_T)}{P \cdot n_T}}$
Probabilistic Case B		No generic model	No generic model
Systematic	Definitely no dependencies	See n-out-of-N	See n-out-of-N
Systematic	Potential dependencies	No generic model	No generic model

Table 6-1: Standard Error for Different Sampling Methods

The accuracy of the sampling schemes depend in different ways on the following parameters:

- the real violator proportion P
- the size of the parent population N
- the target sample size n_T.

From the mathematical models for the estimation accuracy one can observe that the relative standard error for n-out-of-N sampling is equal to the relative standard error for probabilistic sampling multiplied by a factor $\sqrt{(1-P)}$.

$$StdErr_{nofN} = \sqrt{(1-P)} \cdot StdErr_{prob} \tag{6.31}$$

Since $0 \le \sqrt{(1-P)} \le 1$, one can deduce that n-out-of-N sampling provides a smaller standard error and with this a better accuracy than probabilistic sampling. Nevertheless, the difference depends on the violator proportions in the measurement interval and can get very small if there are only few violators.

6.6.1 Dependency on Real Violator Proportion

Figure 6-1 shows how the theoretical standard error (absolute and the relative) depend on the violator proportion *P*. The diagrams show the dependencies for a parent population with N=10,000 packets and different sample sizes n_T.

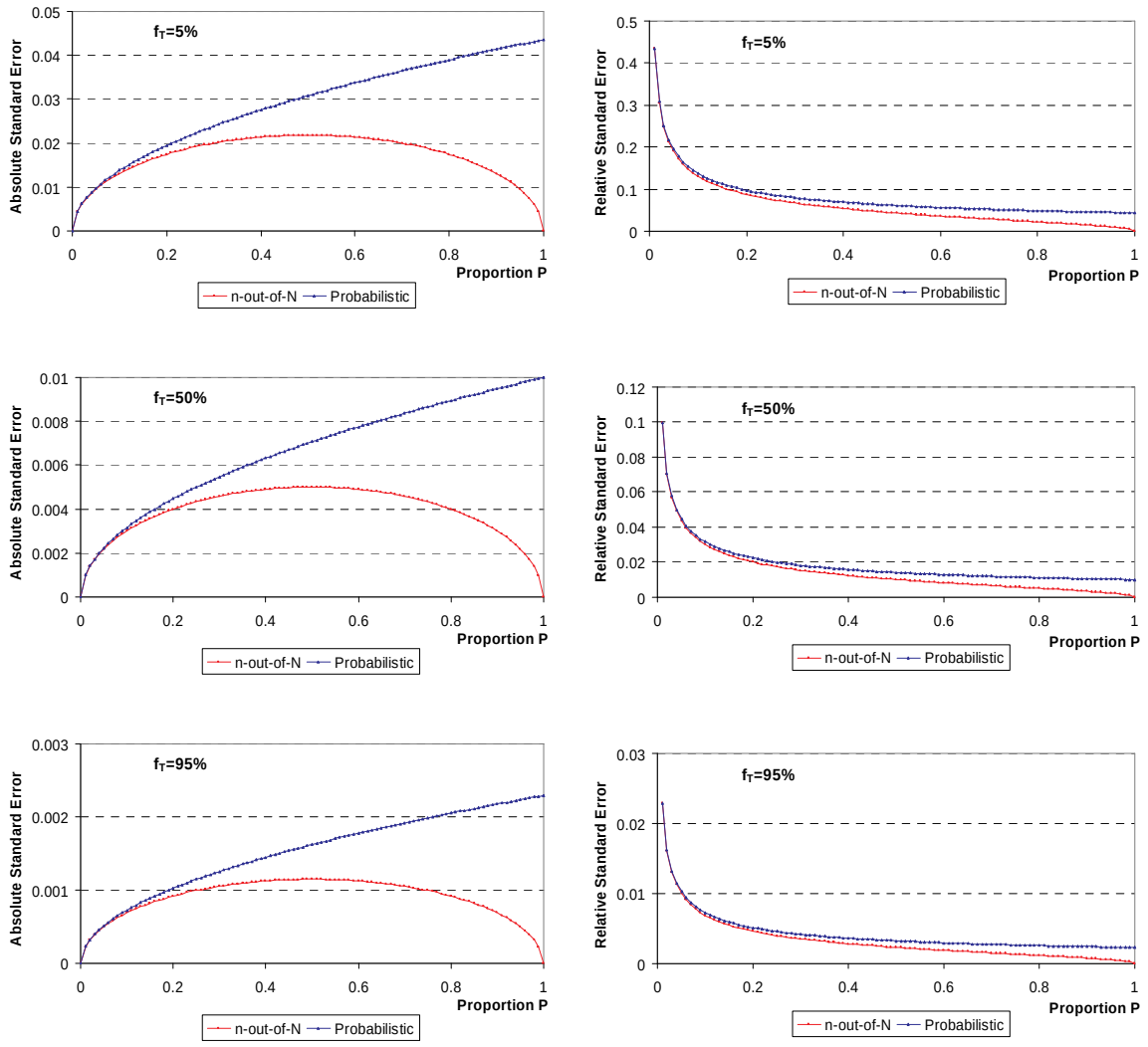


Figure 6-1: Dependency of Absolute and Relative Standard Error on Real Violator Proportion (N=10,000 , n_T=500, 5000, and 9500)

A explained above, the theoretical standard error for probabilistic sampling is always higher than the standard error for n-out-of-N sampling.

For n-out-of-N sampling the absolute error has its maximum at *P=0.5*. For *P=0* and *P=1* the absolute standard error is zero. If all packets are conformant, one always selects n_T conformant packets in each sample run. The estimated violator proportion is $\hat{P}_r = 0$ for each run *r* and therefore the variation of the estimates is zero. The same happens if *P=1*, i.e., if all selected packets are violators, all estimates are 1 and the variation of estimates is zero. If there are only few violators one gets in many runs an estimate $\hat{P}_r = 0$ and in some runs an estimate $\hat{P}_r > 0$. For a proportion of *P=0.5* one gets the highest variation of estimates because half of the packets are violators and half are conformant. The relative standard error decreases for higher proportion because small absolute variations have a smaller effect on them.

For probabilistic sampling the absolute error increases if the proportion increases. The curve differs from the n-out-of-N model. For probabilistic sampling the number of selected packets n_R varies. This is not reflected in the model. So it is assumed that extrapolation is done with the target sample size n_T. If the violator proportion is *P=0*, all selected packets are conformant. So the number of selected violators *m* is always zero. For each sample run the estimated proportion would be $\hat{P}_r = \frac{m}{n_T} = 0$, regardless of the number n_R of packets selected. If the violator proportion is 1, every selected packet is a violator. So the number *m* of selected packets is equal to the number of selected packets n_R. Therefore one gets

$$\hat{P}_r = \frac{m}{n_T} = \frac{n_R}{n_T} \text{ for } P = 1 \tag{6.32}$$

Since n_R differs for each run, one gets a different estimate for each run and the variance of the estimates is higher than zero. So for probabilistic sampling the standard error at *P=1* should equal the standard error of n_R/n_T. With the expected variance of n_R as calculated in section 6.4.2.3 one easily can deduce the absolute standard error of n_R/n_T:

$$StdErr_{abs}\left[\frac{n_R}{n_T}\right] = \sqrt{V\left[\frac{n_R}{n_T}\right]} = \sqrt{\frac{1}{n_T^2} \cdot V\left[n_R\right]} = \sqrt{\frac{1}{n_T^2} \cdot n_T \cdot \left(1 - \frac{n_T}{N}\right)} = \sqrt{\frac{1}{n_T} \cdot \left(1 - \frac{n_T}{N}\right)} \tag{6.33}$$

So one gets the following values which equal the absolute standard errors for *P=1*. Since *P=1*, the relative standard error here is the same as the absolute standard error.

Sample Fraction	**Expected Standard Error of n_R/n_T for P=1**
5%	0.0436
50%	0.01
95%	0.0023

Table 6-2: Expected Standard Error for of n_R/n_T for P=1

The variance of the estimates increases if the proportion increases, because $0 \le m \le M$ and therefore the variability of *m* increases for a higher number *M* of violators in the measurement interval. If one extrapolates with n_R one would expect that the standard error gets to 0 for a

violator proportion *P=1* as for n-out-of-N sampling. The relative error decreases, but never reaches 0 due to the same effect. Furthermore, as expected the absolute and the relative standard error decreases for higher sample fractions f_T for both schemes.

6.6.2 Dependency on Target Sample Size

Figure 6-2 shows how the theoretical standard error depends on the target sample size. The population size *N* was set to *N=10,000*. The target sample size n_T was varied from 100 to 10,000, so that the sample fraction f_T varies from 1-100%. The diagrams show the relative standard error for violator proportions *P=0.01*, *0.5*, *0.99* and *1*.

Since *P* is constant per diagram, the absolute standard error can simply be derived by multiplication of the relative standard error with the constant *P*. That means the shapes of the curves for the absolute standard error look the same.

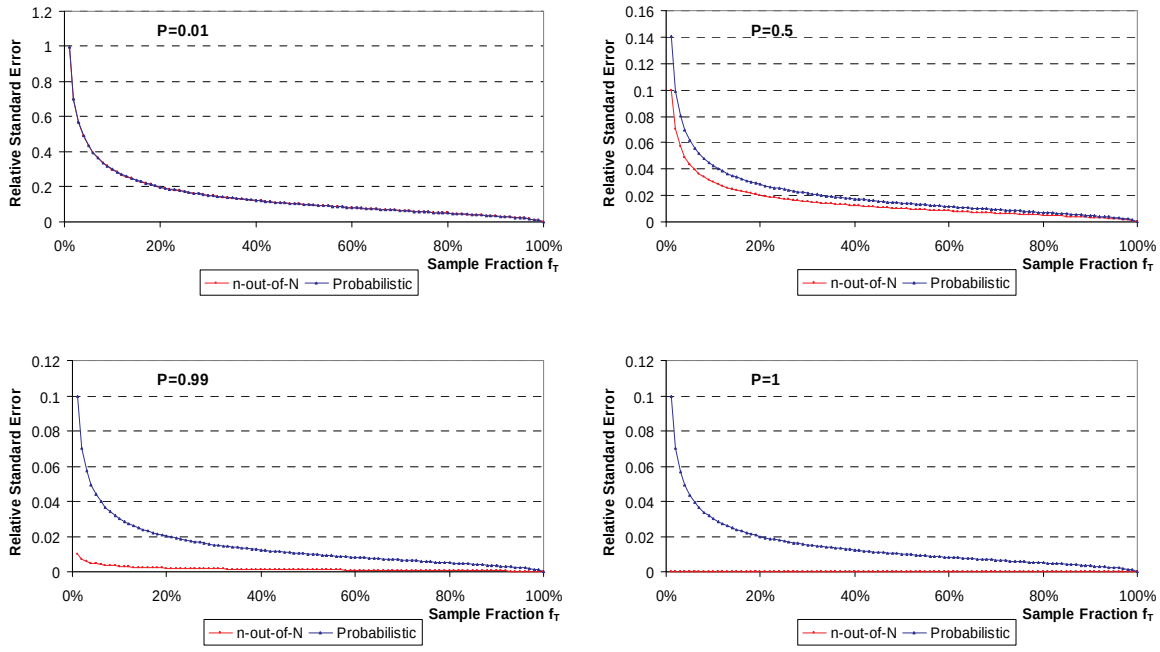


Figure 6-2: Dependency of Relative Standard Error on Target Sample Fraction (N=10,000, Different P)

The standard error decreases for larger sample fractions and approaches zero if the whole population is sampled (f_T=100%) for both schemes. For *P=0.01* the curves for n-out-of-N sampling and probabilistic sampling look similar. For higher proportions the curves diverge. If P increases, the relative standard error decreases (see also Figure 6-1). For *P=1* the standard error for n-out-of-N sampling is 0. For probabilistic sampling with *P=1* we get the standard error for n_R/n_T (see 6.6.1).

6.6.3 Dependency on Measurement Interval Length

The target sample fraction f_T and the violator proportion *P* depend on the number of packets N in the measurement interval.

$$P = \frac{M}{N} \tag{6.34}$$

$$f_T = \frac{n_T}{N} \tag{6.35}$$

Therefore the following behavior is expected:

- If N increases and M remains constant, P decreases and the standard error will evolve as shown in section 6.6.1.
- If N increases and P remains constant, M has to increase too
- If N increases and n_T remains constant, f_T decreases and the standard error will evolve as shown in section 6.6.2.
- If N increases and f_T remains constant, n_T has to increase too

6.7 Proportion Estimation with Stratified Sampling

Stratified sampling methods are explained in section 3.4. Here it is shown how the proportion of non-conformant packets can be estimated by using stratified sampling. The proportion of hits in the sample in stratum l is given by the number of hits m_l in stratum l and the number of all samples n_l that belong to stratum l.

$$p_l = \frac{m_l}{n_l} \tag{6.36}$$

With the weighted sum of the proportions of hits in the sample for each stratum one can calculate an estimate for the proportion of hits in the parent population.

$$\hat{P}_{strat} = \sum_{l=1}^{L} \frac{N_l}{N} p_l \tag{6.37}$$

For stratified sampling the variance of a proportion estimate is calculated as follows (see formula 5.43 in [Coch72]):

$$V_{strat}\left[\hat{P}\right] = \frac{1}{N^2} \cdot \sum_{l=1}^{L} \frac{N_l^2 \cdot \left(N_l - n_l\right) \cdot P_l \cdot \left(1 - P_l\right)}{\left(N_l - 1\right) \cdot n_l} \tag{6.38}$$

If a proportional allocation is used the number of samples per stratum is proportional to the number of elements in the stratum (see 3.4.4). With this formula (6.38) can be simplified to:

$$V_{strat}\left[\hat{P}\right] = \frac{\left(N - n\right)}{N^2 \cdot n} \cdot \sum_{l=1}^{L} \frac{N_l^2 \cdot P_l \cdot \left(1 - P_l\right)}{\left(N_l - 1\right)} \tag{6.39}$$

The standard error is given by the square root of the variance

$$StdErr_{strat}\left[\hat{P}\right] = \sqrt{\frac{\left(N - n\right)}{N^2 \cdot n} \cdot \sum_{l=1}^{L} \frac{N_l^2 \cdot P_l \cdot \left(1 - P_l\right)}{\left(N_l - 1\right)}} \tag{6.40}$$

The standard error for stratified sampling depends on the number N_l of packets per stratum l and on the proportion P_l of violators in that stratum.

The stratification gain Δ_V is defined as the difference between the variance of the estimate that one would get with random sampling V_{rand} and the variance that could be achieved with stratified sampling V_{strat} [Coch72].

$$\Delta_V = V_{rand} - V_{strat} \tag{6.41}$$

The gain Δ_{SE} which shows the differences between the standard errors is defined as follows.

$$\Delta_{SE} = StdErr_{rand} - StdErr_{strat} \tag{6.42}$$

6.8 Prediction of the Estimation Accuracy and Parameter Adaptation

When applying sampling methods, it is crucial to provide information about the expected estimation accuracy in order to inform customers about the degree of potential estimation errors. Nevertheless, Table 6-1 shows that the estimation accuracy depends on the sampling rate *n*, the measurement interval length *N* and the real proportion *P*. The sampling parameters *n* and *N* can be configured. But the real proportion *P* is a traffic parameter that is unknown and usually varies over different intervals. Therefore one needs to approximate the estimation accuracy with parameters known before or after the sampling process.

As shown above, one expresses the estimation accuracy by the absolute standard error. In the following the formulas for n-out-of-N sampling are used because it is the most generic scheme. It is unaffected by potential correlations and is expected to be more accurate than probabilistic sampling.

As already explained in 4.9, there are different ways to approximate the standard error of the estimate:

6.8.1 Approximation with Theoretical Considerations

One can approximate the standard error using 0.25, the maximum value for *P(1-P)*:

$$StdErr_{abs} = \sqrt{\frac{P \cdot (1-P)}{n_T}} \cdot \sqrt{(1-f_T)} \approx \sqrt{\frac{0.25}{n_T}} \cdot \sqrt{(1-f_T)} = \frac{1}{2 \cdot \sqrt{n_T}} \cdot \sqrt{(1-f_T)} \tag{6.43}$$

This approximation is independent of any real value and can be used before the sampling process to estimate the expected accuracy. Nevertheless, this maximum value may be much higher than the real standard error, leading to the assumption of a much too low estimation accuracy.

6.8.2 Estimation from Actual Sampled Values

Another possibility is to use the estimation of P from the sampled packets to calculate the expected accuracy in the measurement interval. With this one gets a more accurate approximation than with just using the maximum value. The estimate from the i^{th} measurement interval is denoted by $\hat{P}_i$.

$$StdErr_{abs} = \sqrt{\frac{P \cdot (1-P)}{n_T}} \cdot \sqrt{(1-f_T)} \approx \sqrt{\frac{\hat{P}_i \cdot (1-\hat{P}_i)}{n_T}} \cdot \sqrt{(1-f_T)} \quad (6.44)$$

With (6.44) the achieved accuracy can only be computed after the sampling process, because only then the characteristics of the sampled packets are known, which are required to calculate $\hat{P}_i$.

6.8.3 Prediction from Previous Samples

If one wants to predict the accuracy before the sampling process, one can estimate *P* by using sample values from a previous measurement interval and a prediction function. The predicted proportion for the i^{th} measurement interval is denoted by P_i'.

$$StdErr_{abs} = \sqrt{\frac{P \cdot (1-P)}{n_T}} \cdot \sqrt{(1-f_T)} \approx \sqrt{\frac{P_i' \cdot (1-P_i')}{n_T}} \cdot \sqrt{(1-f_T)} \quad (6.45)$$

The prediction of the proportion of the actual measurement interval can be based on the estimate of the directly preceding measurement interval only or on multiple previous measurement intervals.

$$P_i' = f\left(\hat{P}_{i-1}\right) \quad (6.46)$$

$$P_i' = f\left(\hat{P}_{i-1}, \hat{P}_{i-2}, \hat{P}_{i-3}, \ldots,\right) \quad (6.47)$$

With this there are two errors that need to be considered. First, one makes an estimation error, when estimating P_{i-1} by $\hat{P}_{i-1}$ from the samples in measurement interval *i-1*. Second, one gets a prediction error when P_i' is predicted from $\hat{P}_{i-1}$.

A similar problem is addressed in [ChPZ02] for measuring traffic load by using an autoregressive (AR) model. It is shown that one has to consider both errors (from prediction and estimation). The usability of this method highly depends on reliable methods to predict the actual proportion from previous measurement intervals. This is not trivial and it is likely that the violator proportion is too dynamic too allow a good prediction. A prediction of violator proportions for real traces is investigated in 7.6.

6.8.4 Adaptive Sampling

The estimation accuracy for estimating the proportion of violators depends on the proportion itself. In order to keep the estimation accuracy stable one could adapt sampling parameters (sample size *n* and measurement interval length *N*) to the expected proportion in the measurement interval. Figure 6-3 shows the involved processes for adapting the sampling parameters. First the proportion is estimated from the sampled packets. Based on this estimation one performs a prediction. The predicted proportion provides the basis to calculate the sampling parameters for the next measurement interval, which then are configured for the sampling process.

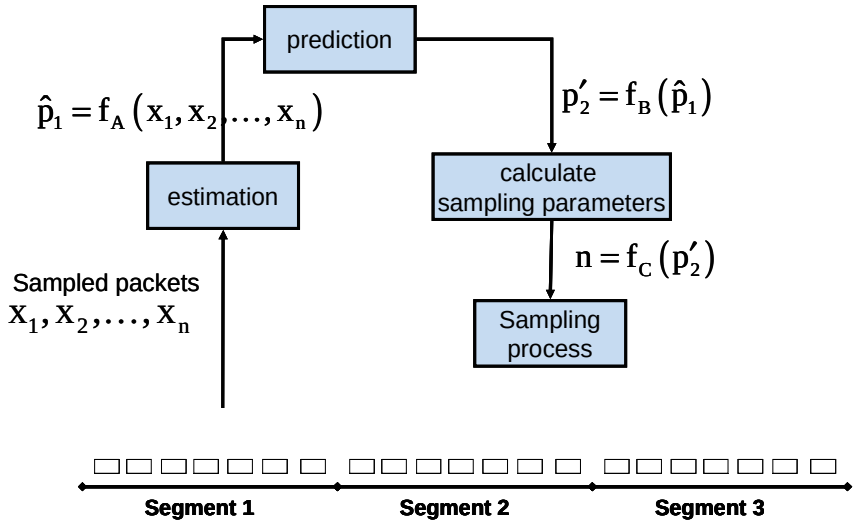


Figure 6-3: Adaptive Sampling

Since one needs to adapt the sampling parameters before the measurement interval starts, one needs to predict the proportion of the measurement interval from previous measurement intervals (see 4.9.3). So the possibility to apply adaptive sampling and with this the stability of the estimation accuracy depends on the quality of the prediction. It is questionable whether the violator proportion is stationary enough too allow a good prediction.

7 Experiments for Violator Proportion Estimation

This chapter describes the results of the empirical investigations for the SLA validation scenario. First the questions are presented that should be answered by experiments, followed by a short description of the software used. The investigated traces are described and the results of the initial trace analysis are shown. After this the sampling experiments and their results are presented and interpreted.

7.1 Questions for Empirical Investigations

As shown in chapter 6, bias and estimation accuracy depend on traffic characteristics. Therefore one part of the empirical investigations is to check what traffic characteristics are found in real traces. With this one can assess what accuracy can be achieved with the different schemes in reality. It also needs to be checked whether empirical results conform to the predicted behavior from the models. Furthermore, it was investigated to what degree a prediction of the estimation accuracy is possible with different methods and whether the deployment of stratified sampling would make sense for the investigated traces. The following questions are investigated by empirical investigations:

- Investigation of traffic characteristics
  - How does the distribution of delay values look like?
    - Are there different delays at all?
    - How far do the delay values spread?
    - Are there many outliers?
    - Do delay distributions from different traces look similar?
  - Are there periodicities or other correlations in the sequence of delay values?
    - Are subsequent delay values correlated?
    - Does the correlation remain if delay values are classified into conformant and non-conformant packets?
- Theoretically Expected vs. Empirical Estimation Accuracy
  - Is the empirical accuracy for the random schemes close to the theoretical values predicted by the models?
  - What accuracy can be achieved with systematic sampling?
  - Does probabilistic sampling perform better or worse than n-out-of-N sampling?
  - Does probabilistic sampling perform better if the extrapolation is done with the real sample size (PROB-R)?
- Accuracy prediction

- Is there a trend in the sequence of violator proportions for subsequent measurement intervals that would allow a good prediction of traffic parameters?
- What is the influence on the standard error if it is calculated with estimated or predicted traffic parameters instead of the real traffic parameters?
- How does the approximated accuracy differ from the statistically expected accuracy (e.g., is the expected confidence level achieved), if the confidence limits are derived from an estimated or predicted standard error?

▪ Stratification
- Is there a suitable stratification variable with high correlation to the survey variable?
- What stratification gain can be expected?

7.2 Analysis Software

For the analysis of the delay traces the sampling simulation library described in section 5.2 is used together with an analysis program that reads in the traces and processes the results. One-way delay is used as an example metric for the empirical investigation. The delay traces are stored in comma separated lists in ASCII format. They contain the arrival time of the packet at the first and the second observation point, the packet size and the calculated one-way delay. The sampling scheme, the threshold *t*, the length of the measurement intervals *N*, the target sample size n_T and the number of runs *R* can be configured with a configuration file.

The analysis program splits the delay trace into measurement intervals. The packets then are classified into conformant and non-conformant packets in accordance to the predefined threshold. It performs a complete analysis of the trace in order to get the real proportion of violators for each measurement interval in the trace. Then the program performs multiple sampling runs over each measurement interval. An estimate for the proportion of violators for each run is computed. The expectation and standard error of the estimate is calculated from these estimates over all runs.

For probabilistic sampling the program allows two options. The estimate can be calculated with the real sample size (PROB-R) or with the target sample size (PROB-T) (see section 5.2.1.1.2). If the estimate is calculated with the target sample size it can happen that the estimate gets larger than $1 \hat{P} > 1$. Since it is clear that the maximum proportion is *P=1*, this knowledge can be used to improve the performance of the PROB-T sampling. Therefore the proportion estimates is limited to $\hat{P} \leq 1$ by setting the estimate to *P=1* if an estimate $\hat{P} > 1$ is observed in the experiments.

Since it is much more likely to observe an estimate that is larger 1 for large proportions, the effect of this limiting of the estimates will be mainly seen in experiments with large proportions.

7.3 Traces

For the empirical investigations different traces from passive one-way-delay measurements were used. Figure 7-1 shows how one-way delay measurements between two observation points are performed. Packet ID and timestamps are collected at the clock-synchronized observation points. The delay is calculated by subtracting the timestamps from the different observation points that are associated with the same packet ID.

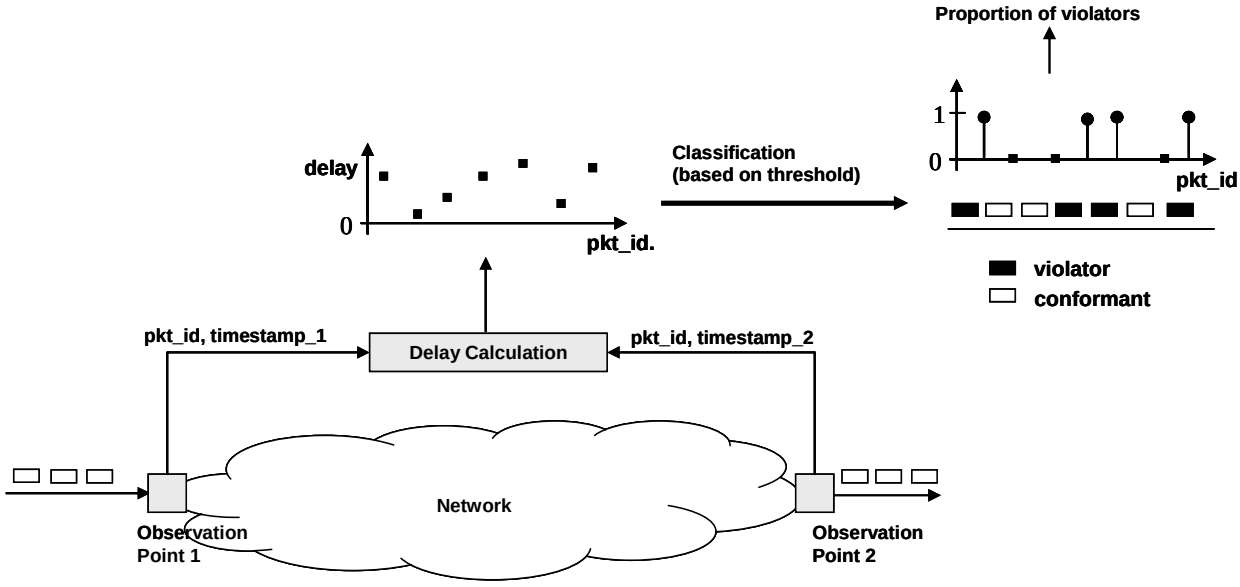


Figure 7-1: One-way Delay Measurements

In the experiments one-way delay traces from video transmissions and distributed gaming were used.

7.3.1 Gaming Traces

Gaming traces were collected during a demo event for IPv6 measurement software developed in the 6QM project [6QM]. Players in Berlin (network A), Madrid (network B) and Kawasaki (network C) participated in a distributed gaming event with Quake2 over IPv6. The involved networks were WIDE (Japan), Euro6IX (Spain) and 6WIN (Germany). GPS and NTP synchronized measurement boxes were installed at all participating locations to perform passive one-way delay measurements between all clients and the server. Further information about the measurements can be found in [DiPP04].

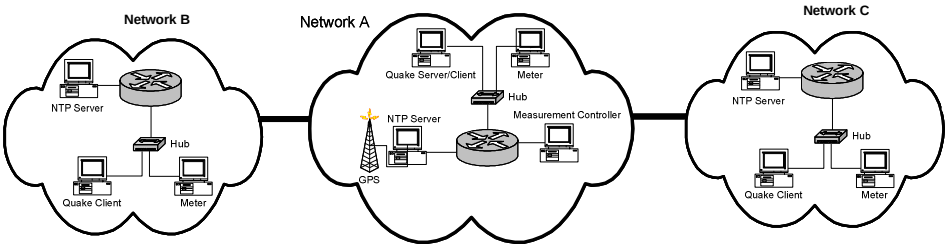


Figure 7-2: Network Configuration (Picture from [6QM] Project)

Figure 7-2 shows the network configuration that was used for measuring the delay of the gaming traffic. The Quake server was hosted in Germany while players were located in Germany, Spain and Japan. Table 7-1 gives an overview of the collected gaming traces.

Trace	Name	Direction	Packets
A	t-1821-1818plus	Berlin (Server) ➔ Kawasaki (Client)	17,979
B	t-1821-1818minus	Kawasaki (Client) ➔ Berlin (Server)	139,756
C	t-1821-1819plus	Berlin (Server)➔ Berlin (Client)	17,971
D	t-1821-1819minus	Berlin (Client)➔ Berlin (Server)	56,628
E	t-1821-1820plus	Berlin (Server)➔ Madrid (Client)	17,900
F	t-1821-1820minus	Madrid (Client)➔ Berlin (Server)	12,714

Table 7-1: Traffic Traces (Quake-II)

7.3.2 Video Traces

Video traces were collected during a video transmission between Germany and Slovakia. The video server was located in Berlin (Germany) and a client in Kosice (Slovakia). Two VCON Cruiser75 videoconferencing systems were used. The system uses [H.261] as video codec and [G.711] as audio codec. Measurement boxes were installed at both sites to perform passive one-way delay measurements. The clocks of the measurement boxes where synchronized using the network time protocol (NTP). The measurement boxes where directly served by the GPS synchronized stratum 1 server. Two different test series where performed. In the first, the movie Matrix was transmitted from the server to the client. In the second there was only a still picture transmitted with the video server. Table 7-2 gives an overview of the collected video traces. A more detailed description of the collected traces can be found in [CoSu03].

Trace	Name	Direction	Packets
G	Vidconf-owd-matrix-Berlin-to-Kosice.dat	Berlin (Server) ➔ Kosice (Movie: Matrix)	85,698
H	Vidconf-owd-matrix-Kosice-to-Berlin.dat	Kosice ➔ Berlin (Server) (Movie: Matrix)	133,834
I	Vidconf-owd-nomovie-Berlin-to-Kosice.dat	Berlin(Server) ➔ Kosice (no movie)	64,798
J	Vidconf-owd-nomovie-Kosice-to-Berlin.dat	Kosice ➔ Berlin (Server) (no movie)	180,979

Table 7-2: Traffic Traces (Video)

7.4 Traffic Characteristics

This section contains an analysis of the traffic characteristics of the measured traces. It shows the traffic parameters that influence the estimation accuracy and is used as basis for selecting traces for the sampling simulation.

7.4.1 Delay Distributions

Figure 7-3 shows the boxplots of the delay values for the gaming traces. Figure 7-4 contains the boxplots for the video traces. Each boxplot shows minimum, maximum, median, 1st and 3rd quartile of the delay values in the trace and all delay values that lie outside those values. The length of the box is from the 1st to the 3rd quartile and represents the variability of the observations. The median is indicated by a line with notches. The whiskers end at 1.5 of the inter quartile range (IQR), so all packets outside the range of the whiskers can be considered as outliers. The lengths of the vertical lines at each boxplot correspond to the square root of the size of the population, i.e., the square root of the number of packets in the trace. Therefore the large trace B, H, and J have longer vertical lines.

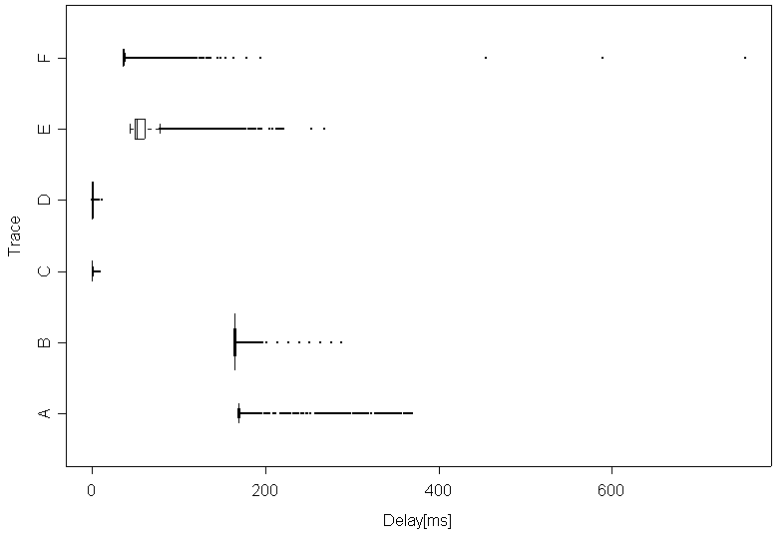


Figure 7-3: Boxplots of Delay in Gaming Traces (Whole Traces)

For most delay traces the boxes are so small that one cannot distinguish between the median and the quartiles. That means in general the delay values do not spread very much (see also Table 7-3). Nevertheless, for the traces A, E and F one can see a large amount of delay values that are larger than the 3rd quartile and also beyond 1.5 of the inter quartile range. As expected the delay between the client in Berlin and the server, which also was located in Berlin, (traces C and D) is pretty small and has also only small variations. Therefore those traces are not so interesting for the investigations. As expected the highest mean delay can be observed for the traces between Berlin and Kawasaki (A, B).

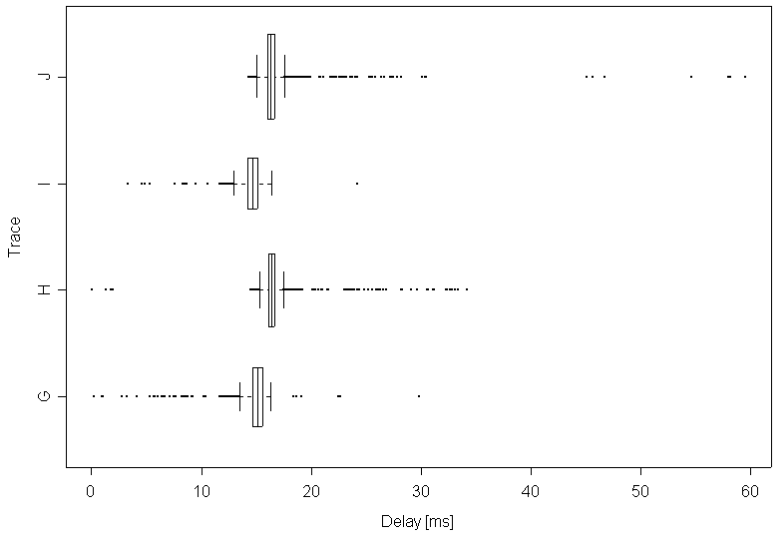


Figure 7-4: Boxplots of Delay in Video Traces (Whole Traces)

The delay values for the video traces are all in the same range. This is not surprising because the same Internet connection was used for all video experiments. More outliers (values above 1.5 IQR) are observed for the packets from server to client (traces H and J) than for the

packets from client to server. Furthermore, the mean delay for packets that travel from Berlin to Kosice (traces I and G) is a little bit smaller than for packets that travel the reverse path (see Table 7-3). There are only few differences between the delay distributions of the transmission of the video Matrix (traces G and H) and transmission of the transmission of a still picture (traces I and J).

Trace	Min.	1st Quartile	Median	3rd Quartile	Max.	Mean	StdDev
A	168.10	168.90	169.10	169.50	369.60	170.80	13.87
B	163.60	164.10	164.30	164.80	288.00	164.70	1.32
C	0.001	0.13	0.19	0.26	8.80	0.20	0.14
D	0.00	0.32	0.39	0.45	11.20	0.39	0.13
E	43.58	49.19	52.00	60.79	268.20	59.20	18.80
F	35.45	35.82	36.00	36.41	753.60	38.64	13.11
G	0.27	14.71	15.16	15.54	29.80	15.09	0.58
H	0.08	16.13	16.38	16.66	34.19	16.37	0.49
I	3.29	14.25	14.66	15.11	24.23	14.72	0.60
J	14.30	16.01	16.30	16.64	59.49	16.31	0.55

Table 7-3: Delay Statistics (in [ms]) of All Traces

7.4.2 Packet Size Distributions

Figure 7-5 shows the boxplots of the packet sizes for the gaming traces. The gaming traces contain only packets that are smaller than 150 bytes. Furthermore, all packets in the traces from clients to the server (traces B, D, F) have the same packet size (apart from a few outliers in trace D). For all traces the median has the same value than the 1st quartile (see also Table 7-4). That means that many packets have the same size (the size of the median). The largest packets are observed to the client in Madrid (trace E), the smallest to the client in Kawasaki (trace A).

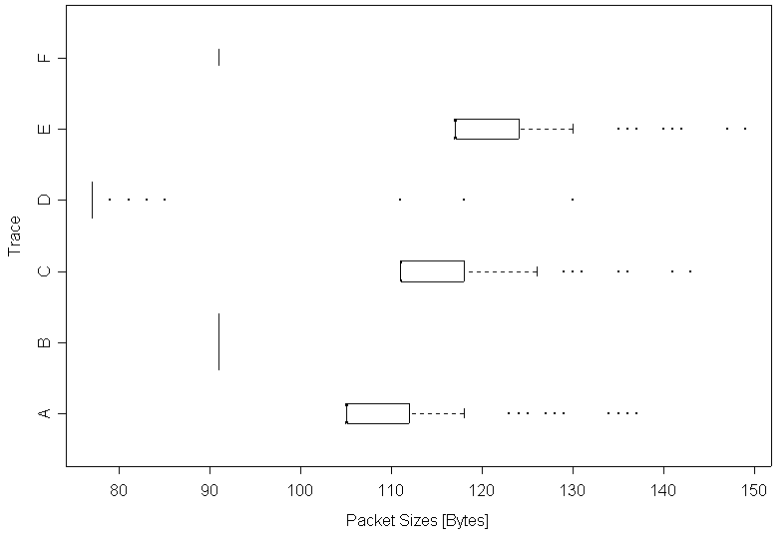


Figure 7-5: Boxplots of Packet Sizes in Gaming Traces (Whole Traces)

Figure 7-6 shows the boxplots for packet sizes in the video traces. In contrast to the gaming traces here packets of all sizes (0 to 1500) are observed and the distributions are very broad. Only for trace I outliers can be observed. In all other traces all packet sizes lie within 1.5 of the IQR, because the IQR is so large, that is spans the whole range of possible size values. In trace I the median is equal to the 1st quartile (at 520 bytes). That means many packets of the same size (520 bytes) are observed. All other distributions are skewed slightly to the right, which means large packets are more frequent than small packets.

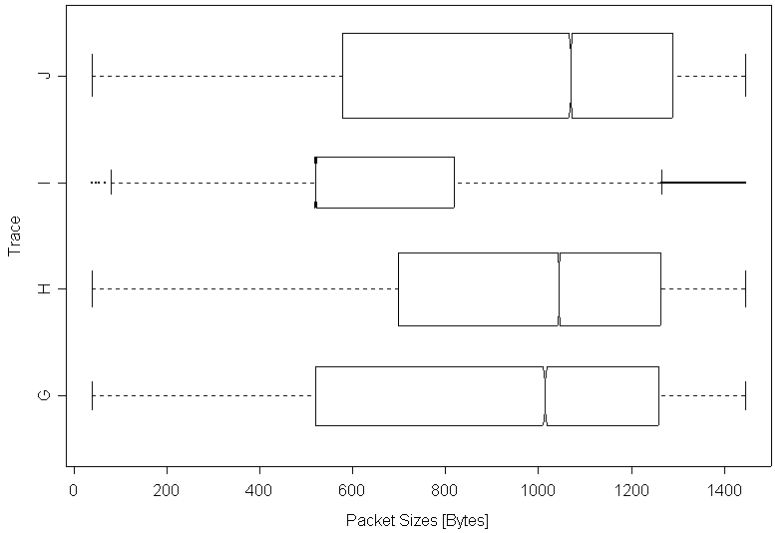


Figure 7-6: Boxplots of Packet Sizes in Video Traces (Whole Traces)

Trace	Min	1st Quartile	Median	3rd Quartile	Max	Mean	StdDev	Coeff. of Variation
A	105	105	105	112	137	108.9	4.719187	0.04333505
B	91	91	91	91	91	91	0	0
C	111	111	111	118	143	113.7	3.922715	0.03450057
D	77	77	77	77	130	77.02	0.8662188	0.01124667
E	117	117	117	124	149	120.2	4.046019	0.03366072
F	91	91	91	91	91	91	0	0
G	40	520	1014	1259	1445	940.6	360.8062	0.38359154
H	40	699	1044	1264	1445	975.5	339.0895	0.34760584
I	40	520	520	818	1445	682.1	335.7569	0.49223999
J	40	579	1069	1289	1445	974.6	359.0436	0.36840099

Table 7-4: Packet Size Statistics (in [Bytes]) of All Traces

7.4.3 Autocorrelation of Delay Values

Figure 7-7 shows the autocorrelation functions for the gaming traces. In all traces there is at least a small autocorrelation, of the observed delay values. In trace A the highest correlation coefficient for lags >0 is observed for lag=2 (0.372). For lags higher than 4 the correlation coefficient gets close to 0.

In trace B quite large correlation for small lags can be seen. Trace C and D show a nearly constant autocorrelation factor for all lags shown. The value is around 0.25 for trace C and around 0.29 for trace D. Traces E and F show only very few autocorrelation. In both traces the highest value (for lags > 0) is observed at lag=1 (0.134 for trace E and 0.298 for trace F). In trace F additionally high (but decreasing) values are observed for lags that are multiple of 9.

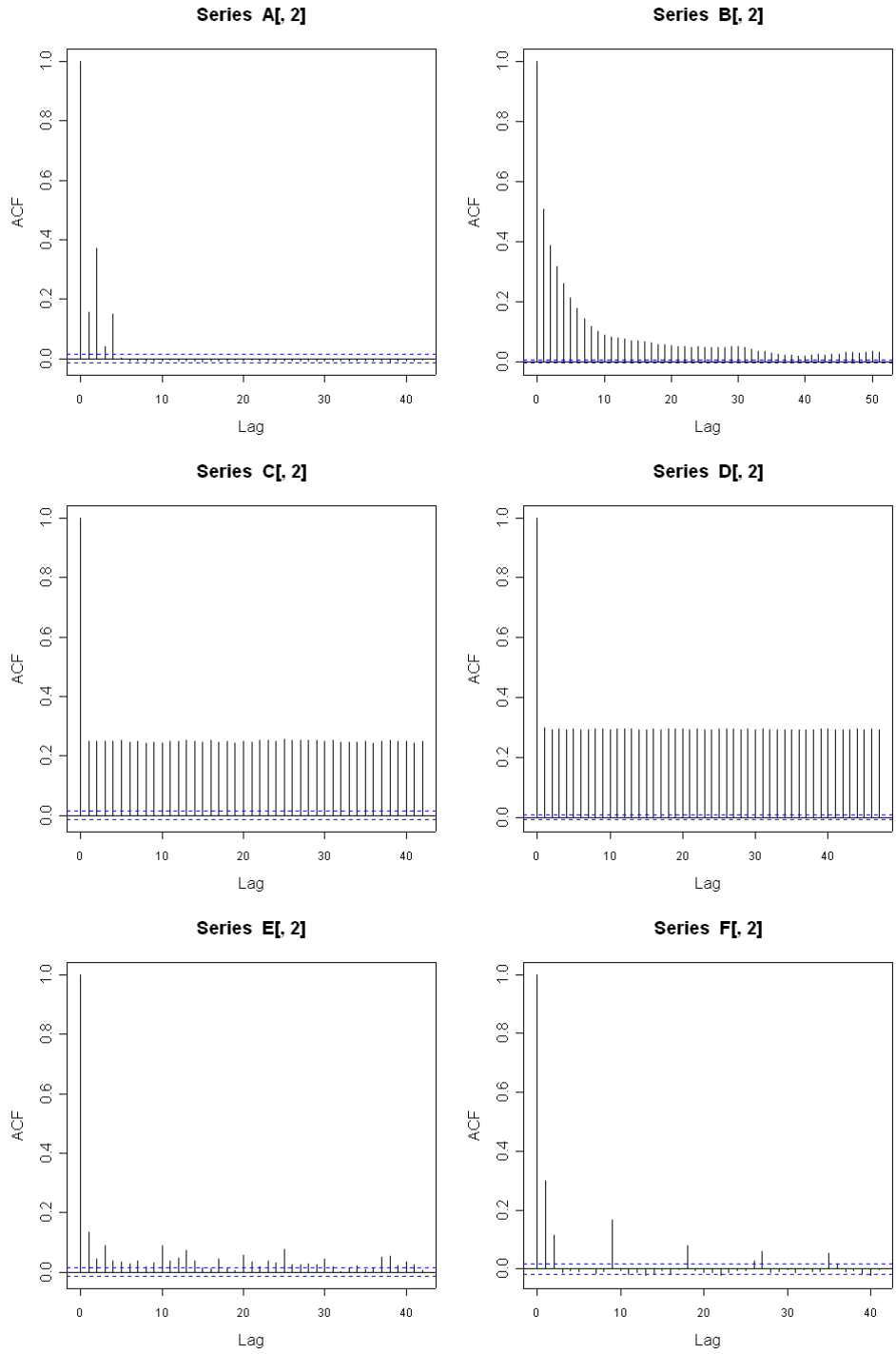


Figure 7-7: Autocorrelation of Delays in Gaming Traces

Figure 7-8 shows the autocorrelation functions for the video traces. Trace G shows nearly no correlation. Trace H shows only a few correlations. For both traces the highest value for lags>0 is observed at lag=1 (0.063 for trace G, 0.153 for trace H). Traces I and J show much larger correlations (0.217 and 0.246 for lag=1) that stays large also for larger lags.

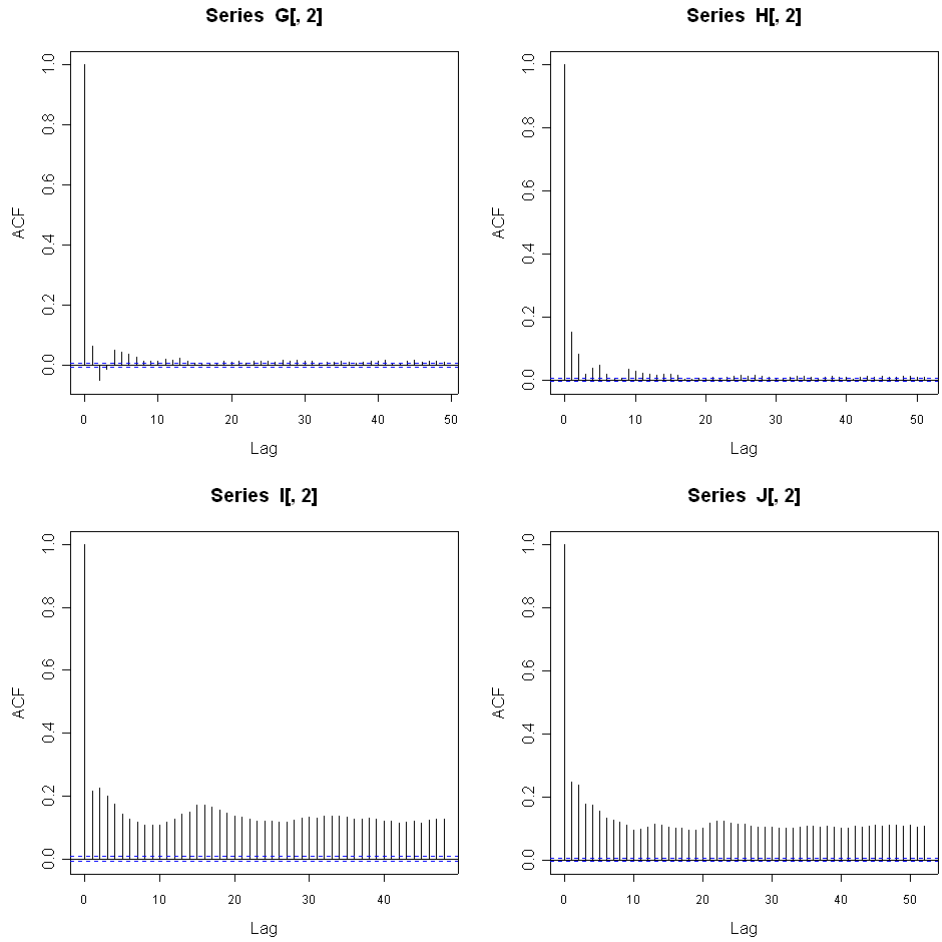


Figure 7-8: Autocorrelation of Delays in Video Traces

After analyzing the correlations of the delay values it was investigated what correlations occur after the classification of packets into conformant and non-conformant packets. Different delay thresholds were used for the classification. In all cases the correlation was very small. This is true for different thresholds. That means that the conformance or non-conformance of subsequent packets is not or only slightly correlated. Therefore no significant effects are expected due to correlations and it is assumed that systematic sampling performs similar to random schemes.

7.4.4 Measurement Intervals

For the experiments the delay traces are split into measurement intervals of 10,000 packets. That means for each trace one gets multiple measurement intervals. The analysis is done only on complete measurement intervals, which contain exactly 10,000 packets. Remaining packets at the end of the trace that are to few to form a further complete measurement interval are not considered.

Trace	Packets	Measurement Intervals
A	17,979	1
B	139,756	13
C	17,971	1
D	56,628	5
E	17,900	1
F	12,714	1
G	85,698	8
H	133,834	13
I	64,798	6
J	180,979	18

Table 7-5: Complete Measurement Intervals in Traffic Traces

In order to work with traces with small and higher correlation, the traces B, E, G, and I are selected for further experiments.

7.4.5 Classification

Packets are classified into conformant and non-conformant packets in accordance to a pre-defined threshold. After this, one gets a series of 0/1 values for each trace. In order to investigate the influence of different violator proportions to the accuracy of the sampling methods different delay thresholds are used for the classification. In order to get the same violator proportion in all investigated measurement intervals, the delay threshold is set to the percentiles of the delay distribution. Table 7-6 shows the percentiles of the delay distributions of the selected measurement intervals.

Trace	1^{st}	5^{th}	25^{th}	50^{th}	75^{th}	95^{th}	99^{th}
B-mi1	163.7830	163.8750	164.0740	164.2910	164.8500	166.5232	168.0512
E-mi1	45.09390	46.43595	49.09350	51.91050	60.31150	95.95240	130.49940
G-mi1	14.02299	14.10100	14.77300	15.22500	15.59400	15.90705	16.00100
I-mi1	13.95598	14.17400	14.46900	14.89400	15.34600	16.05900	16.24100

Table 7-6: Percentiles (in ms) of Delay Distributions (First Measurement Intervals)

That means if the delay threshold for trace B-mi1 is set to the value of the 1^{st} percentile (163.783 ms) one gets a violator proportion of 99%. If the threshold is set to the 99^{th} percentile (168.0512 ms) the violator proportion is 1%.

For the experiments proportions of 1%, 50% and 99% violators are used. Also the autocorrelation functions of the 0/1 series were generated as explained in 7.4.3. But only for trace B-mi1 a few correlations are observed. For all other traces the occurrence of conformant and non-conformant packets was uncorrelated.

7.5 Bias and Precision

Selected measurement intervals from the traces are investigated to compare the effects of different sampling methods for different traces. By using single measurement intervals it is ensured that the same number of packets is investigated from each trace. Furthermore the delay thresholds can be adjusted in a way to get equal violator proportions for the investigated measurement intervals.

The purpose of these experiments is to investigate whether the bias and accuracy predicted by the models is also observed in reality and to check whether any effects can be observed if systematic sampling is used. Furthermore, it is investigated whether a difference can be observed between the video and the gaming traces and how different violator proportions affect the estimation accuracy.

For this purpose different measurement intervals are analyzed: the first measurement interval (first 10,000 packets) of trace B (gaming, some delay correlation), trace E (gaming, low delay correlation), trace G (video, low delay correlation) and trace I (video, some delay correlation). Although the delay correlation differs for those traces none or only very small correlations exists after classifying the packets into conformant and non-conformant packets. Therefore it is expected that also bias and accuracy for systematic sampling are close to the theoretical model for random sampling.

7.5.1 Experiment Description

The traces B-mi1, E-Mi1, G-mi1 and I-mi1 contain the first measurement interval (N=10,000 packets) of the traces B, E, G, and I. For the empirical investigation of bias and precision R=10,000 sample runs are performed on these sub-traces. Three different thresholds are used to realize violator proportions of 0.01, 0.5 and 0.99 for each trace Tests are performed with different sample fractions (1%-100%) and different sampling methods. Table 7-7 summarizes the experiment settings

Input File	Sampling Methods	Threshold [µs]	Violator Proportion	Sample Fractions
B-mi1	BITSET, PROB-R, PROB-T, SYSTEMATIC	168051.2	0.01	1% - 100%
B-mi1	BITSET, PROB-R, PROB-T, SYSTEMATIC	164291	0.501	1% - 100%
B-mi1	BITSET, PROB-R, PROB-T, SYSTEMATIC	163783	0.99	1% - 100%
E-mi1	BITSET, PROB-R, PROB-T, SYSTEMATIC	130499.4	0.01	1% - 100%
E-mi1	BITSET, PROB-R, PROB-T, SYSTEMATIC	51910.5	0.5	1% - 100%
E-mi1	BITSET, PROB-R, PROB-T, SYSTEMATIC	45093.9	0.99	1% - 100%
G-mi1	BITSET, PROB-R, PROB-T, SYSTEMATIC	16001	0.0102	1% - 100%
G-mi1	BITSET, PROB-R, PROB-T, SYSTEMATIC	15225	0.5002	1% - 100%
G-mi1	BITSET, PROB-R, PROB-T, SYSTEMATIC	14022.99	0.99	1% - 100%
I-mi1	BITSET, PROB-R, PROB-T, SYSTEMATIC	16241	0.0101	1% - 100%
I-mi1	BITSET, PROB-R, PROB-T, SYSTEMATIC	14894	0.501	1% - 100%
I-mi1	BITSET, PROB-R, PROB-T, SYSTEMATIC	13955.98	0.99	1% - 100%

Table 7-7: Experiments Overview

7.5.2 Estimation Errors

Figure 7-9 shows the violin plots of the estimation errors for the four investigated traces. Violin plots are extended boxplots, which show, in addition to median and quartiles, the approximated distribution of the estimation errors.

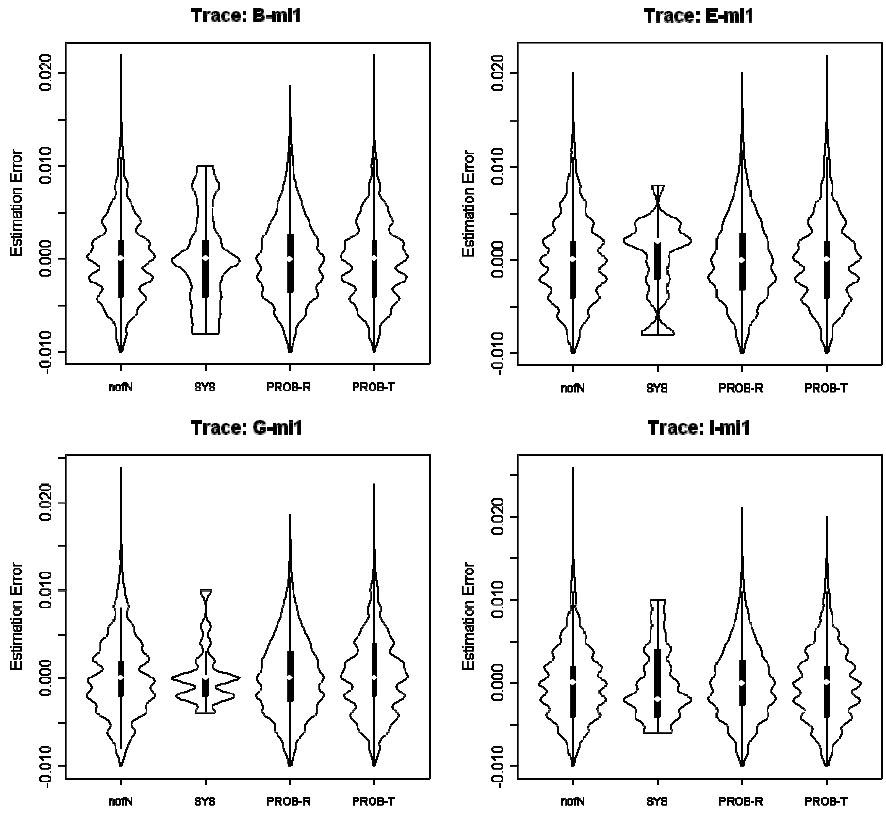


Figure 7-9: Violin Plots of Estimation Errors for Traces B-mi1, E-mi1, G-mi1 and I-mi1, Sample Fraction=5%, P=0.01, Different Sampling Methods

For the random schemes (nofN, PROB-T, PROB-R), the distributions of the estimation errors look similar, quite symmetric and close to a normal distribution. Only the distribution for systematic sampling differs and looks multimodal. This can be explained by the fact that with random sampling all combinations of *n* packets out of all *N* packets in the population are possible, whereas with systematic sampling the selection is limited because only packets with the same distance can be selected.

Plots for other sample fractions look similar. For higher sample fractions the distribution shapes of the random schemes further approach a normal distribution, whereas the distribution for systematic sampling still shows multiple modes (see Figure 7-10). For f_T=50% (K=2) only two different estimates can occur with systematic sampling. Therefore the histogram has only two peaks. For violator proportions P=0.5 one gets larger estimation errors for all schemes. With P=0.99 much lager estimation errors occur with PROB-T sampling than with the other schemes (see Figure 7-11). Figure 7-11 also shows the limitation of the estimation error for PROB-T sampling to 0.01 for P=0.99, caused by the limitation of the estimates to $\hat{P} \leq 1$ for PROB-T sampling (see 7.2).

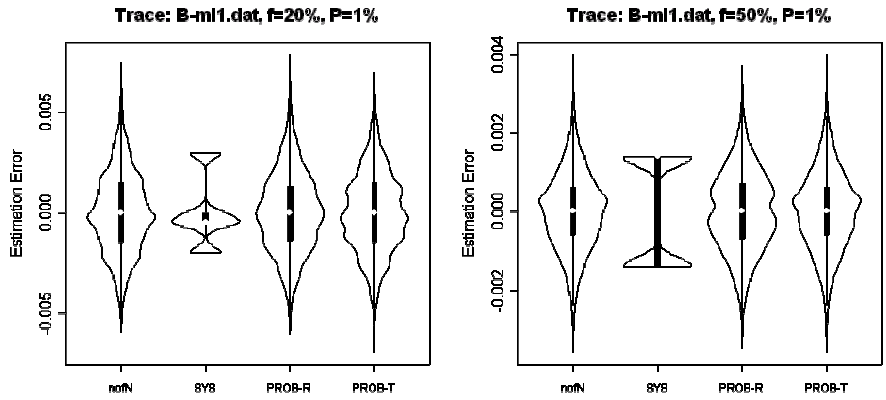


Figure 7-10: Violin Plots of Estimation Errors for Traces B-mi1, Sample Fraction 20% and 50%, P=0.01, Different Sampling Methods

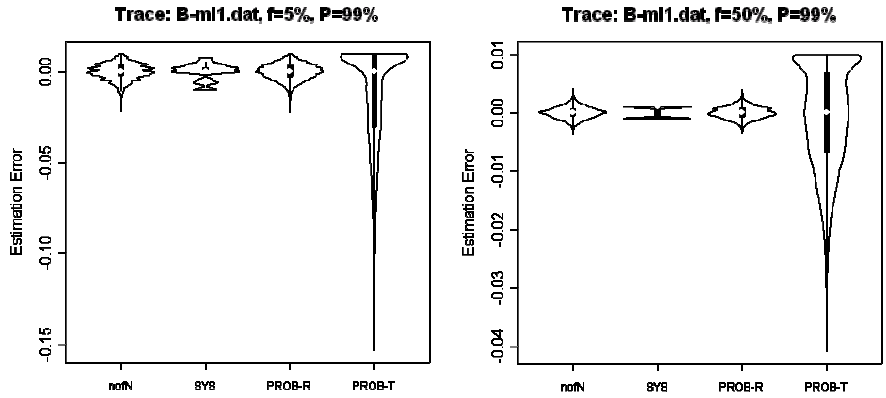


Figure 7-11: Violin Plots of Estimation Errors for Traces B-mi1, Sample Fraction=5%, P= 0.5 and 0.99, Different Sampling Methods

7.5.3 Variability of Sample Size for Probabilistic Sampling

For probabilistic sampling the real sample size varies and can differ from the target sample size. According to theory the distribution of n_R approaches a binomial distribution (see section 6.4.2.3). Figure 7-12 shows this effect for the experiments with 10,000 runs on trace B-mi1 for target sample fractions of 5% and 95%.

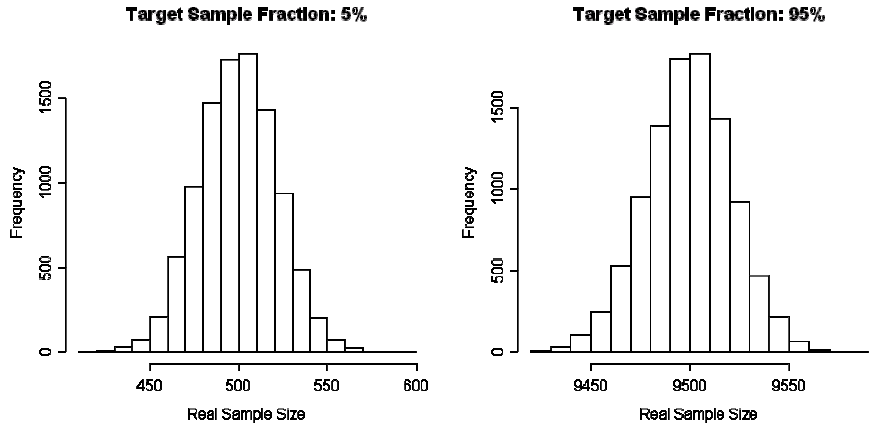


Figure 7-12: Histograms of Sample Size for Target Sample Fraction=5% and 95 %, (PROB-R, Trace B-mi1, P=0.01)

Table 7-8 shows the expected mean and variance for n_R, calculated with formula (6.30) in section 6.4.2.3) compared to the mean and variance for n_R observed in the experiments with 10,000 sample runs. The variance of n_R has its maximum at $f_T=0.5$. The values observed in the experiments are close to the theoretically expected values.

Sample Fraction	Expected Mean	Observed Mean	Expected Variance	Observed Variance
5%	500	500.00	475	472.82
50%	5000	4999.91	2500	2526.35
95%	9500	9499.83	475	471.32

Table 7-8: Expected and Observed Mean and Variances for Real Sample Size

7.5.4 Bias

The empirical bias is calculated as difference between the mean of the estimates (from all R runs) and the real value.

$$Bias_{abs,emp} = \frac{1}{R} \cdot \sum_{r=1}^{R} \hat{P}_r - P \qquad (7.1)$$

Figure 7-13 shows the bias for the investigated traces for the different schemes. The bias decreases if the sample fraction increases. For all schemes the bias is very small. One cannot observe a difference for the traces with no correlation (E-mi1 and G-mi1) compared to traces with some correlation (F-mi1 and I-mi1). This can be explained by the observation that the correlations nearly vanish after the classification into conformant and non-conformant packets (see 7.4.3). The diagrams show the target sample fraction on the x-axis. Please note that for probabilistic and systematic sampling the real sample fraction can differ from the target sample fraction. For sample fractions >50%, the bias for systematic sampling is zero, because the sampling period *K* is 1 and therefore the real sample fraction is 100 %, i.e., all packets are selected.

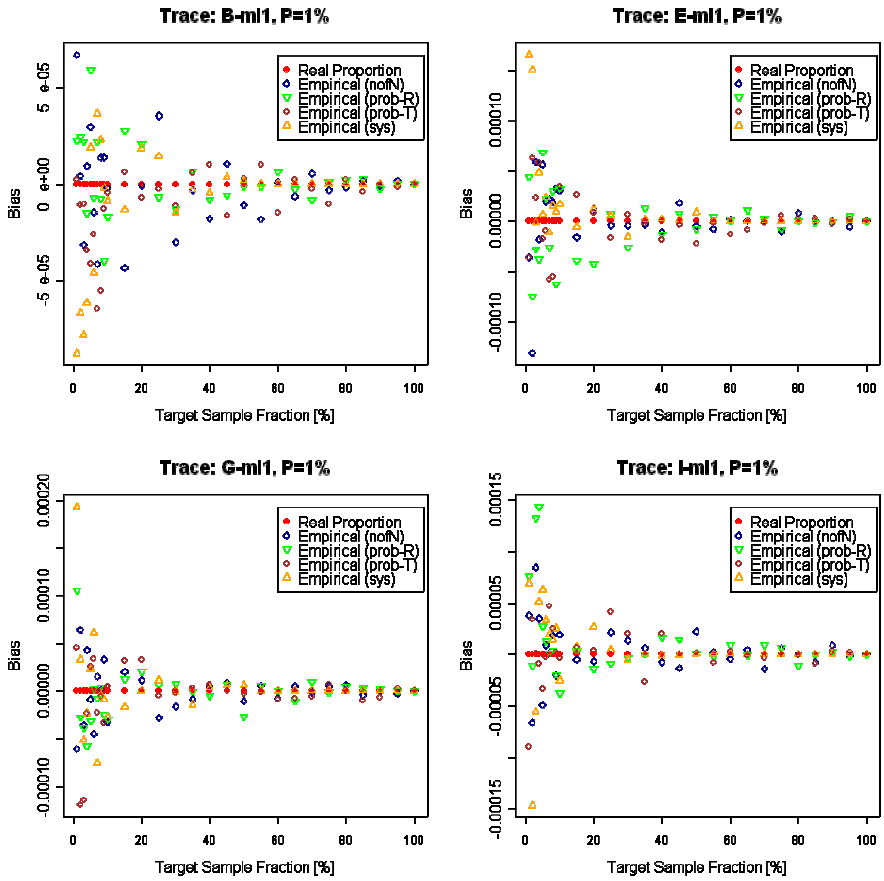


Figure 7-13: Bias for Traces B-mi1 E-mi1,G-mi1 and I-mi1, P=0.01, Different Sample Fractions, Different Sampling Methods

Figure 7-14 shows the bias for the estimates from experiments with trace B-mi1 with different violator proportions (realized by different delay thresholds). Again only a very small bias is observed. For higher proportions the bias for probabilistic sampling with extrapolation with n_T (PROB-T) increases. Due to the limitation to $\hat{P} \leq 1$ for PROB-T sampling, one gets more estimates below the real value than above. Therefore a comparatively high negative bias is observed for small sample fractions. The results for the other traces for higher violator proportions look similar.

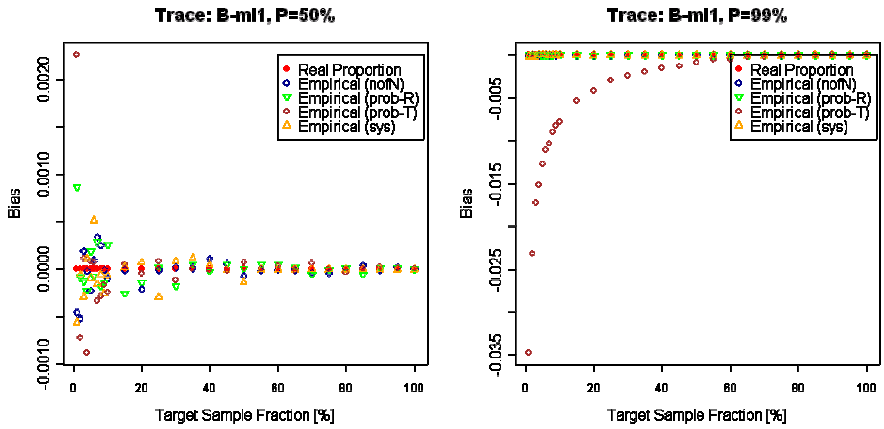


Figure 7-14: Bias for Trace B-mi1, P=0.5 and 0.99, Different Sample Fractions, Different Sampling Methods

7.5.5 Precision

In order to compare values from different traces, the precision is expressed by the relative standard error of the estimates. The empirical standard error is calculated from the estimates $\hat{P}_r$ of the runs as follows:

$$StdErr_{rel,emp}\left[\hat{P}\right] = \frac{\sqrt{V\left[\hat{P}\right]}}{P} = \frac{\sqrt{\frac{1}{R}\cdot\sum_{r=1}^{R}\left(\hat{P}_r - P\right)^2}}{P} \tag{7.2}$$

In the experiments the empirical standard error is calculated from the estimates from all runs. Figure 7-15 shows the relative standard error for the investigated traces for a violator proportion of *P=0.01* and different schemes. The lower the standard error of the estimate the higher is the precision. Figure 7-16 shows the relative standard error for trace B-mi1 for different violator proportions. The diagrams show the target sample fraction on the x-axis. Please note that the real sample fraction can differ from the target sample fraction for systematic and probabilistic sampling (see section 5.2.1.1).

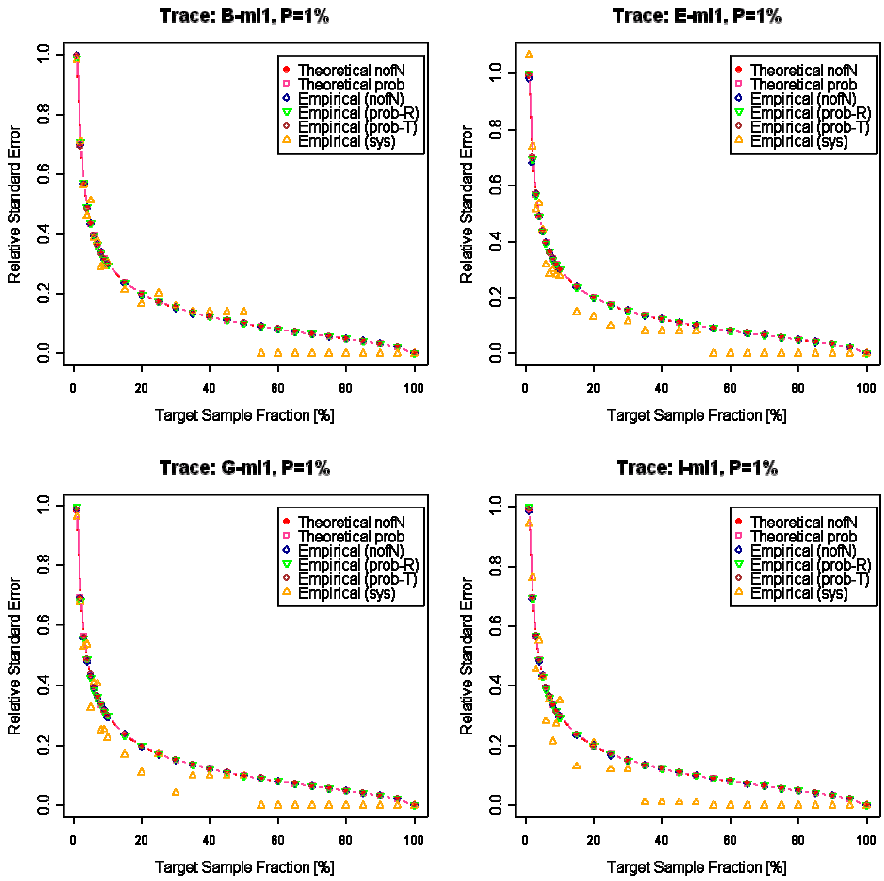


Figure 7-15: Relative Standard Error for Traces B-mi1, E-mi1, G-mi1 and I-mi1, P=0.01, Different Sample Fractions, Different Sampling Methods

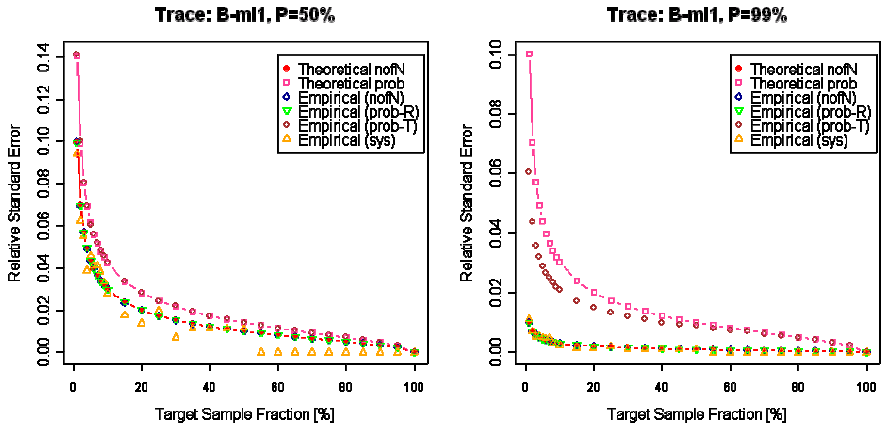


Figure 7-16: Relative Standard Error for Traces B-mi1, P=0.5 and 0.99, Different Sample Fractions, Different Sampling Methods

As expected in 6.6 the theoretical curves for n-out-of-N and probabilistic sampling look equal for small violator proportions and diverge for larger proportions. It can be seen that the accuracy for n-out-of-N sampling and for PROB-R sampling is better than the PROB-T sampling, especially for large violator proportions.

The empirical results for n-out-of-N sampling are close to the n-out-of-N model. The empirical results for probabilistic sampling with extrapolation with the target sample size n_T (PROB-T) match the values predicted by the probabilistic model for small proportions. For the proportion *P=0.99* one get better results than predicted by the model. The smaller standard error is achieved here due to the limitation of the estimates to $\hat{P} \leq 1$. That means by using the a-priori information that $P \leq 1$ one can achieve a higher accuracy. The empirical results for the probabilistic sampling with extrapolation with the real sample size n_R (PROB-R) differ from the probabilistic mode and are close to the n-out-of-N model instead.

Systematic sampling performs in most cases better than random sampling. For target sample fractions above 50% the period for the systematic sampling is *K=1*, i.e., all packets are selected. Therefore the standard error is 0 for systematic sampling with sample fractions above 50%.

7.5.6 Conclusion

The trace analysis showed quite different delay distributions for gaming and video traces. The distributions are all quite narrow but for some connections several outliers (values above 1.5 times the IQR) are observed, which are potential SLA violators. The autocorrelation functions of the traces showed some correlations for a few traces if delay values are analyzed. Nevertheless, after classifying the delay values into conformant and non-conformant the correlations decline to a minimum.

The sampling experiments showed that the bias for all schemes is very small. Even with systematic sampling the estimates look unbiased. As expected the bias decreases for larger sample fractions. The distribution of the estimation errors approaches a normal distribution for the random schemes, whereas multi-modal distributions were observed for systematic sampling.

The empirical precisions for the n-out-of-N sampling and the PROB-T sampling are very close to values predicted by the mathematical models derived in 6.3 and 6.4. Only for large proportions the empirical PROB-T precision is better than predicted by the model, because the a-priori knowledge that $P \leq 1$ was used in order to limit the estimates to $\hat{P} \leq 1$. For small proportions the PROB-T results look similar to the n-out-of-N results but for large violator proportions the PROB-T sampling performed worse. The perceived estimation errors and bias was larger than for all other schemes

The empirical results of the PROB-R sampling (extrapolation with real sample size n_R) are close to the n-out-of-N model. The observed variation of the real sample size n_R for probabilistic sampling is in accordance to the expected theoretical values. Furthermore, one gets a much higher precision for probabilistic sampling if the real sample size n_R can be used for the extrapolation instead of the target sample size n_T, especially for high violator proportions. That means a higher accuracy can be gained if an additional counter is provided to count the number of selected packets.

For QoS measurements sometimes multipoint measurements are needed where synchronization between the sampling schemes at different observation points is required. Such synchronization can be realized by a hash-based selection that emulates probabilistic sampling [DuGr00]. If the uniformity of the distribution is ensured similar results are expected as for the probabilistic sampling. So for traces with small violator proportions one could use hash-based selection methods with both extrapolation options (PROB-T and PROB-R) and would get nearly the same accuracy as for n-out-of-N sampling.

Systematic sampling performs quite well for the investigated traces. The observed bias was very small and in many cases a higher precision could be achieved than for n-out-of-N sampling. One reason by which the good performance for systematic sampling in the experiments can be explained are the small correlations at small lags and the lack of periodicities in the delay sequences. When using systematic sampling, one avoids selecting subsequent (potentially correlated) packets. If the correlations do not occur periodically with the sampling period, one gets more uncorrelated packets in the sample.

One can use the n-out-of-N model for systematic sampling, if it is clear that all delay values are independent. And it is likely to achieve a better performance if correlations occur at small lags. Nevertheless, since periodicities cannot generally be excluded, the systematic sampling results cannot be generalized for arbitrary traces.

7.6 Prediction of the Estimation Accuracy

As shown in chapter 6 the expected estimation accuracy depends on the proportion of violators itself. But if sampling techniques are applied in a real life environment, the real proportion of violators is unknown before and after the sampling process. Therefore one has to rely on values that are known or gained during the measurements to predict the expected estimation accuracy. In 6.8 different techniques were shown to approximate the estimation accuracy, by using values that are known before or after the sampling process. The following techniques can be used for calculating the standard error to get a prediction of the estimation accuracy:

1. Use the maximum value with[18] $PQ_{\max} = P \cdot (1-P) = 0.25$
2. Use the estimate $\hat{P}$ from sampling results from the actual measurement interval
3. Use a prediction P' from sampling results from previous measurement intervals

In the following it is checked how well the estimation accuracy can be predicted with the different techniques. First the maximum value and the estimation from values from the actual measurement interval are investigated (case 1 and 2). Then it is analyzed how good the estimation accuracy can be assessed if a prediction from values of the previous measurement interval is used (case 3).

[18] The term PQmax is used here, because (1-P) is often referred to as Q.

7.6.1 Prediction from Actual Measurement Interval

First it is investigated how the standard error can be assessed by using known or estimated values from the actual measurement interval. The first measurement interval of trace B (B-mi1) is taken as basis for the experiments. The n-out-of-N model is used, because it is the best model for small violator proportions. Results are shown for different sampling fractions.

7.6.1.1 Predicted Standard Error

The largest absolute standard error is observed if $P \cdot (1-P) = 0.25$ and the smallest if $P \cdot (1-P) = 0$. In the first case one overestimates the standard error and assumes a lower accuracy than one actually would get. In the second case the standard error is underestimated and a higher accuracy than one actually would get is assumed. If the standard error is approximated with estimates from the sample, one gets different standard errors and in some cases overestimates and in some underestimates the real standard error of the proportion estimate.

Figure 7-17 shows the approximated absolute standard error, calculated with the different methods, for the sampling fractions f_T*=1, 2, 4, 5, 10, 20, 25,* and *50*. It shows the results from trace B-mi1 (first measurement interval of trace B) with a real violator proportion of *P=0.01*, *P=0.5* and *P=0.99*. The diagrams show the standard error calculated with the real violator proportion (red solid line), the maximum standard error, calculated with PQ_{max} (blue dashed line) and the distribution of the approximated standard errors, calculated from the estimated proportion of each sampling run (black boxplots).

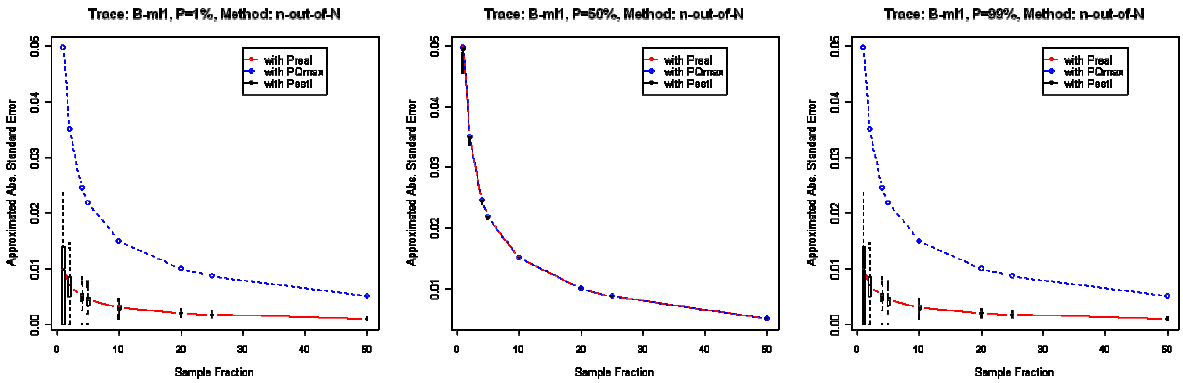


Figure 7-17: Prediction of Absolute Standard Error with Different Methods

The theoretical maximum standard error, calculated with PQ_{max}, is much higher than the real standard error if the proportion is smaller or larger than 0.5. That means the accuracy would be underestimated and a much higher accuracy could be achieved in reality.

The approximated standard errors from the estimated proportions are much closer to the theoretical standard error. That means for *P=0.01* and *P=0.99* one gets a much better approximation if the estimates from the actual measurement interval are used to calculate the

approximated standard error. For larger sampling fractions the approximation of the standard error from the estimated proportion gets closer to the real standard error.

For a real proportion of $P_{real}=0.5$ one gets $P_{real} \cdot (1-P_{real}) = 0.25$. That means the curve for P_{real} lies exactly on the curve for the approximation with PQ_{max}. Also the values from the experiments are equal or close to 0.5. Only for small sample fractions, the minimum values lie a little bit below.

7.6.1.2 Confidence Limits

Since the approximated standard errors can differ from the theoretical standard error calculated with the real proportion it is now investigated what effect this has on the calculation of the confidence limit and on the achieved confidence level.

For each estimate $\hat{P}$ one can formulate a confidence interval. If a normal distribution can be assumed for the estimate, the confidence limits can be calculated with the critical factor z_c as follows:

$$\hat{P} - \varepsilon \le P \le \hat{P} + \varepsilon \tag{7.3}$$

$$\varepsilon = z_c \cdot StdErr_{abs}\left[\hat{P}\right] = z_c \cdot \sqrt{\frac{P \cdot (1-P)}{n_T}} \cdot \sqrt{(1-f_T)} \tag{7.4}$$

If a confidence level of 95% should be achieved, $z_c=1.96$ has to be used. For a confidence level of 99% the boundaries are calculated with $z_c=2.58$.

For each run one gets an new estimate $\hat{P}$ for P. Based on this estimate the confidence limits are calculated. So even if the theoretical standard error (calculated with the real P) was known, one would get different CI limits per run. Statistically one gets in 95% of all runs confidence limits that contain the real value if the limits are calculated for a CI level of 95%. Nevertheless, in reality only the estimate $\hat{P}_i$ (from the sample in the i^{th} measurement interval) is known. If the standard error is calculated based on the estimate $\hat{P}$ instead with the real P, one gets slightly different CI limits.

$$\varepsilon \approx z_c \cdot Approx\left[StdErr_{abs}\left[\hat{P}\right]\right] = z_c \cdot \sqrt{\frac{\hat{P}_i \cdot (1-\hat{P}_i)}{n_T}} \cdot \sqrt{(1-f_T)} \tag{7.5}$$

Figure 7-18 shows the CI limits for the first 1000 runs from a trial with trace B-mi1 with sample fraction $f_T=5\%$. The real violator proportion was $P_{real}=0.01$. The CI limits are calculated for a confidence level of 95%. The left diagram shows the CI limits if the real standard error (calculated with the real proportion P) is used to express ε. The right diagram shows the CI limits if the approximated standard error (calculated with the estimate $\hat{P}$ of each run) is used to express ε. The red dots show the upper CI limit and the blue dots express the lower CI limits. In all cases where a red dot is below the real $P=0.01$ or a blue dot is above $P=0.01$, the real value lies outside the CI limits. Statistically that should only happen in 5% of all cases, because the confidence level was set to 95%.

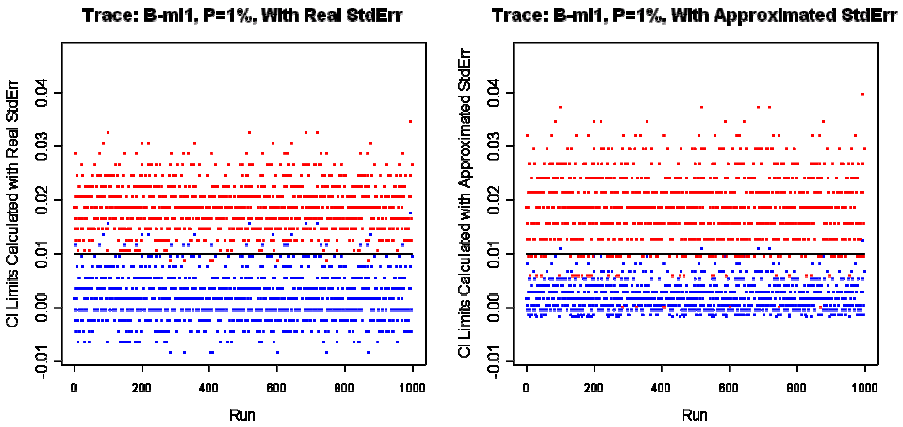


Figure 7-18: CI Limits for Confidence Level 95% with Real and Approximated Standard Error

In 96.41% of all 10,000 runs the real violator proportion lies within the CI limits if the real standard error is taken as basis. If the approximated standard error is used instead for the calculation of the CI limits, the real violator proportion lies only in 88.16 % of all runs within the CI limits. That means that the achieved confidence level is smaller than statistically expected if the approximated standard error is used.

Figure 7-19 shows the percentage of runs (of the 10,000 runs) for which the real proportion lies in the calculated CI limits for different sample fractions. The CI limits per run are calculated by only using the estimated proportion.

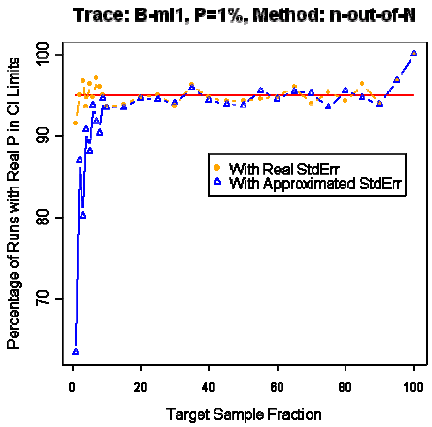


Figure 7-19: Percentage of Runs where Real Proportion is within CI Limits

For small sample fraction the percentage of runs in which the CI limits contain the real value differs from the expected 95% if the approximated standard error is used for the calculation of the CI limits. For a sample fraction of 1% the real proportion lies only in 63.41% of all runs within the calculated CI limits. With the real standard error a percentage of 91.58 % can be achieved. When the sample fraction is increased, the estimates get more accurate and therefore the curve for the CI limits calculated with the approximated standard error also approaches the expected 95%. For a sample fraction of 100% one gets in each run the exact value for the violator proportion and therefore the real value lies in the CI limits in all runs

(100%) for both schemes. For sample fractions >10% the difference between the curves remains below 2%.

Figure 7-20 show the percentage of runs where the real value lies in the CI limits calculated with the approximated standard error for different methods.

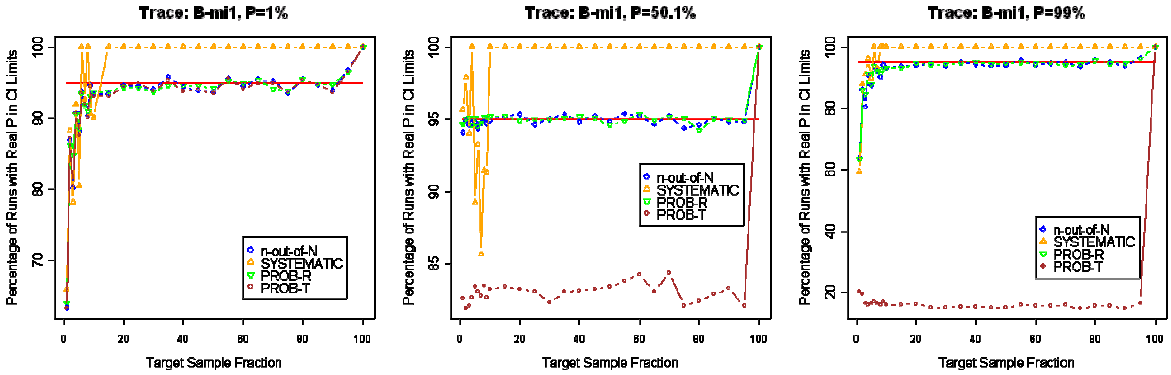


Figure 7-20: Percentage of Runs where Real Proportion is within CI Limits

For small proportions the results from all schemes are close to the theoretical expected confidence level of 95% for sample fractions above 10%. Only the systematic sampling performs sometimes much better. For higher violator proportions the PROB-T scheme performs much worse than the other schemes. This can be explained by the large errors that are made when estimating the proportion by extrapolating with the target sample fraction. With this method one gets bad estimates and therefore the approximated standard error highly differs from the theoretically expected standard error.

7.6.2 Prediction from Previous Measurement Intervals

For the analysis of a series of measurement intervals trace B (13 measurement intervals) and trace G (8 measurement intervals) are selected. A constant threshold of $d_{max}=168051.20\ \mu s$ is used for trace B and $d_{max}=16001\ \mu s$ for trace G to classify the packets into conformant and non-conformant (violator) packets. Since a violation of the SLA should be an exception in a well dimensioned network, rather small violator proportions are expected in reality. Therefore the highest thresholds from the experiments above are used, which result in small violator proportions. Table 7-9 and Figure 7-21 show the proportion of violators per measurement interval for both traces.

Trace B	
Measurement Interval	Proportion
1	0.0100
2	0.0062
3	0.0132
4	0.0117
5	0.0195
6	0.0430
7	0.0171
8	0.0064
9	0.0060
10	0.0130
11	0.0360
12	0.0187
13	0.0151

Trace G	
Measurement Interval	Proportion
1	0.0102
2	0.0115
3	0.0038
4	0.0005
5	0.0014
6	0.0021
7	0.0011
8	0.0002

Table 7-9: Proportions per MI for Traces B and G

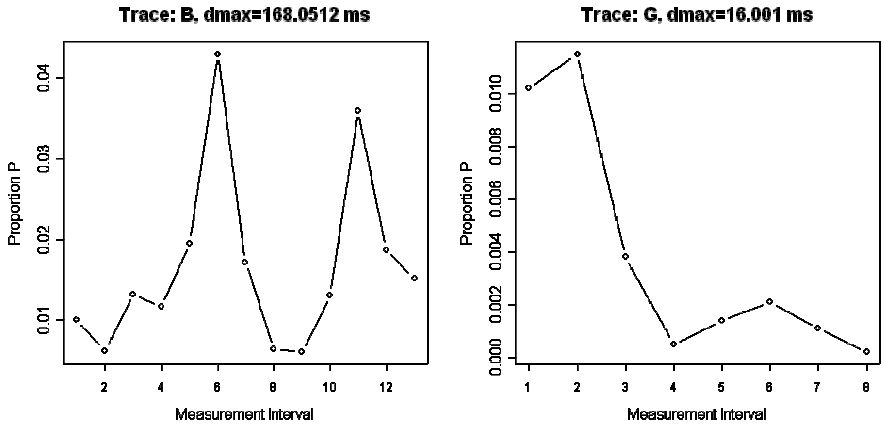


Figure 7-21: Violator Proportion for all Measurement Intervals in Trace B and G (N=10,000)

For trace B two peaks with very high violator proportions are observed. In trace G the violator proportion decreases for higher measurement intervals. For some measurement intervals (4 and 8) the proportion of violators is very small. No general trend can be observed, that would allow a good prediction of the proportion from previous measurement intervals. Therefore it is tested in a subsequent experiment whether more smooth curves evolve if shorter measurement intervals are defined. Figure 7-22 shows the violator proportions for all measurement intervals with the same thresholds as above, but with a shorter measurement interval length of *N=1000* packets.

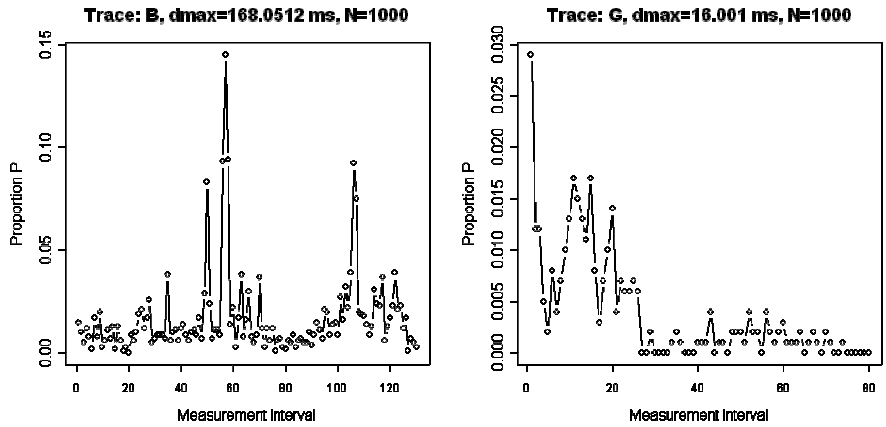


Figure 7-22: Violator Proportion for all Measurement Intervals in Trace B and G (N=1000)

With the more fine grained segmentation, more intermediate values evolve and the proportions seem to change not so rapidly. Nevertheless there are still very high and comparatively sudden peaks. Furthermore, there are several intervals with a violator proportion of 0. In the following experiments it is checked whether a prediction of the standard error from the previous measurement interval is possible and whether the prediction can be improved if smaller measurement intervals are used.

7.6.2.1 Predicted Standard Error

For the investigation of the prediction of the standard error from the previous measurement interval, sampling experiments are performed with all measurement intervals of both traces. No trend could be observed in the sequence of proportions of the different measurement intervals (Figure 7-21, Figure 7-22). Therefore in the experiments only a very simple prediction method is used, i.e., the predicted proportion for the current interval is set to the proportion (real or estimated) of the previous interval.

In these experiments the sample fraction remains constant at 5%. Furthermore, a higher number of runs (*R=100,000*) is used to get more accurate empirical results. First it is compared how the expected standard error per measurement interval differs if it is calculated with different methods. The following methods are compared:

- Calculation of the standard error with the real proportion P_i from the actual measurement interval *i*. Since the real proportion remains the same independent of the sample runs, one gets exactly one standard error per measurement interval.
- Calculation of the standard error with the estimated proportion $\hat{P}_i$ from the actual measurement interval *i*. Since the estimated proportion is different for each sample run, one gets a different standard error per run (for each measurement interval).

- Calculation of the standard error with the real proportion P_{i-1} from the previous measurement interval *i-1*. Since the real proportion remains the same independent of the sample runs, one gets exactly one standard error per measurement interval.
- Calculation of the standard error with the estimated proportion $\hat{P}_{i-1}$ from the previous measurement interval *i-1*. Since the estimated proportion is different for each sample run, one gets a different standard error per run (for each measurement interval).

Figure 7-23 and Figure 7-24 show the expected absolute standard error per measurement interval calculated with different methods for trace B and G and a measurement interval length of *N=10,000*. The left diagrams compare the standard error calculated with the real proportion (red solid line) with the approximated standard errors calculated with the estimated proportion in this measurement interval (black boxplots). Since one gets an estimate per run, and therefore one approximated standard error per run the graph shows the distribution of the approximated standard errors as boxplots.

The right diagrams show the standard error calculated with the real proportion of the previous measurement interval (blue dashed line) and the distribution of the approximated standard errors calculated with the estimated proportion of the previous measurement interval (black boxplots). For comparison it also contains the curve for the theoretically expected standard error (red solid line).

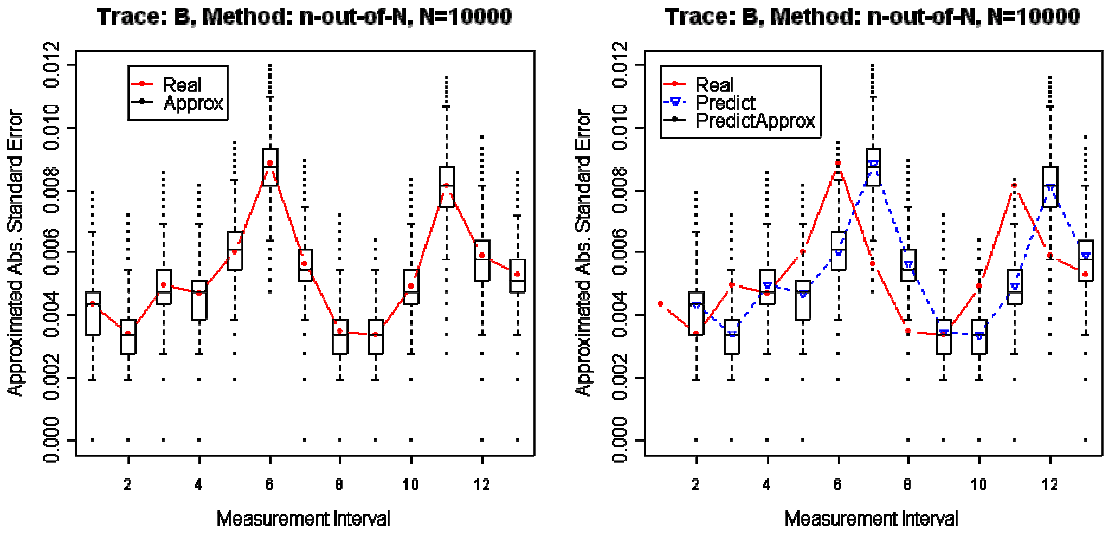


Figure 7-23: Predicted Absolute Standard Error for all Measurement Intervals in Trace B (N=10,000)

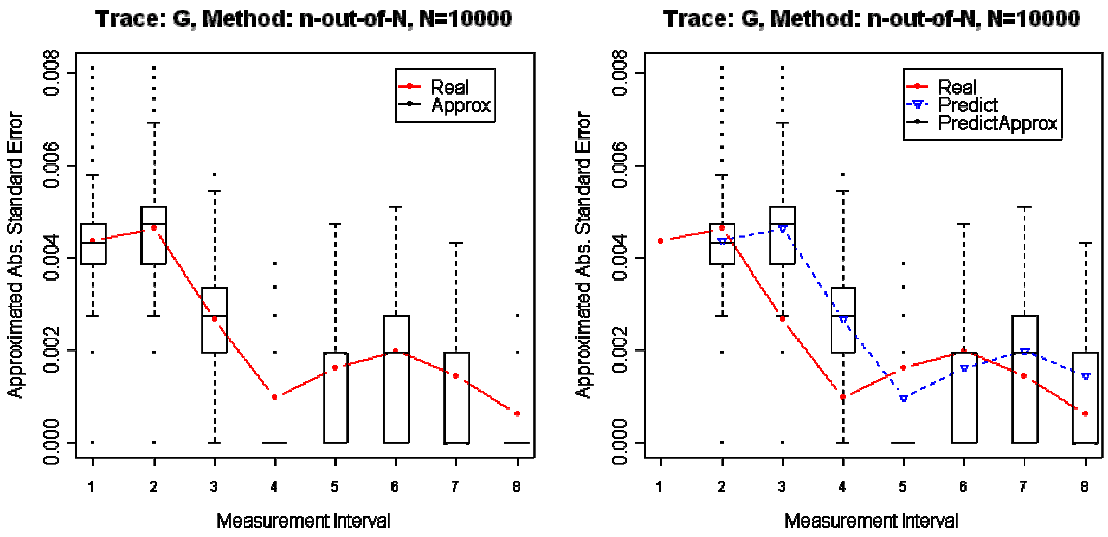


Figure 7-24: Predicted Absolute Standard Error for all Measurement Intervals in Trace G (N=10,000)

One can see that the predicted standard errors can differ extremely from the theoretical standard error calculated with the real proportion. When the standard error is calculated from the estimates of each run, one gets a wide range in which the estimated standard errors can lie. In most cases the minimum standard error is 0. This happens, because at least in one of the 100,000 runs the estimate is $\hat{P} = 0$. If a proportion of $P=0$ is assumed, the expected standard error is 0, because if all packets conform, there can be no variation in the estimates. If the real proportion from the last measurement interval is used to predict the standard error, it highly depends on the similarity of subsequent proportions, whether the predicted value is close to the real value or not. Nevertheless, for all traces and all measurement intervals the standard error predicted from previous intervals is closer to the real standard error than the maximum

standard error calculated from current estimates. With the predicted values from estimates from previous intervals, one again gets a wide range of possible standard errors.

Figure 7-25 and Figure 7-26 show the results for trace B and G if smaller measurement intervals are used (*N=1000* packets). Since the presentation of all boxplots would be too dense for such a large number of measurement intervals, here only the maxima (pink line) and minima (green line) of the approximated standard errors are shown instead of the whole distributions.

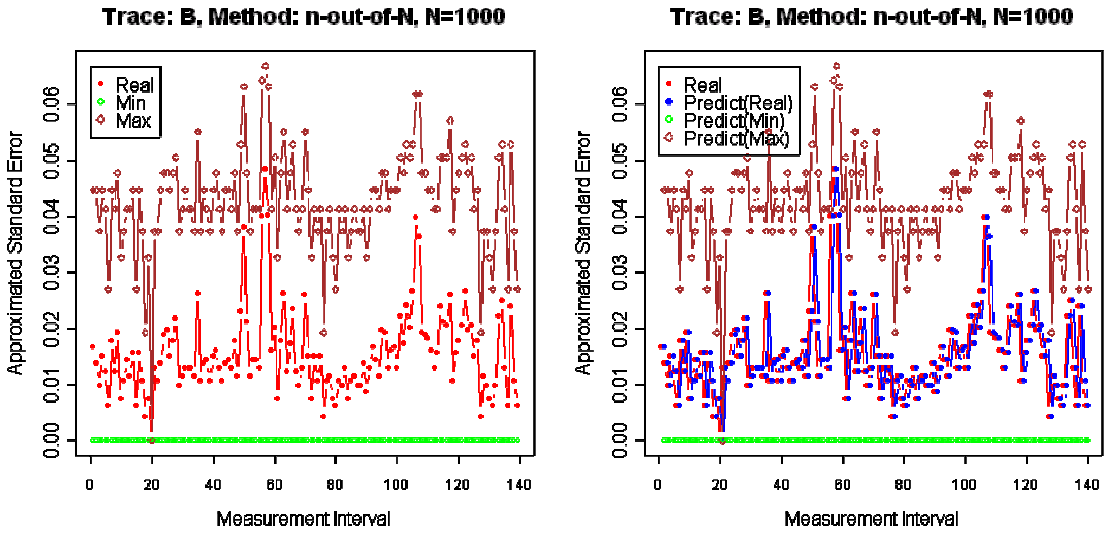


Figure 7-25: Predicted Absolute Standard Error for all Measurement Intervals in Trace B (N=1000)

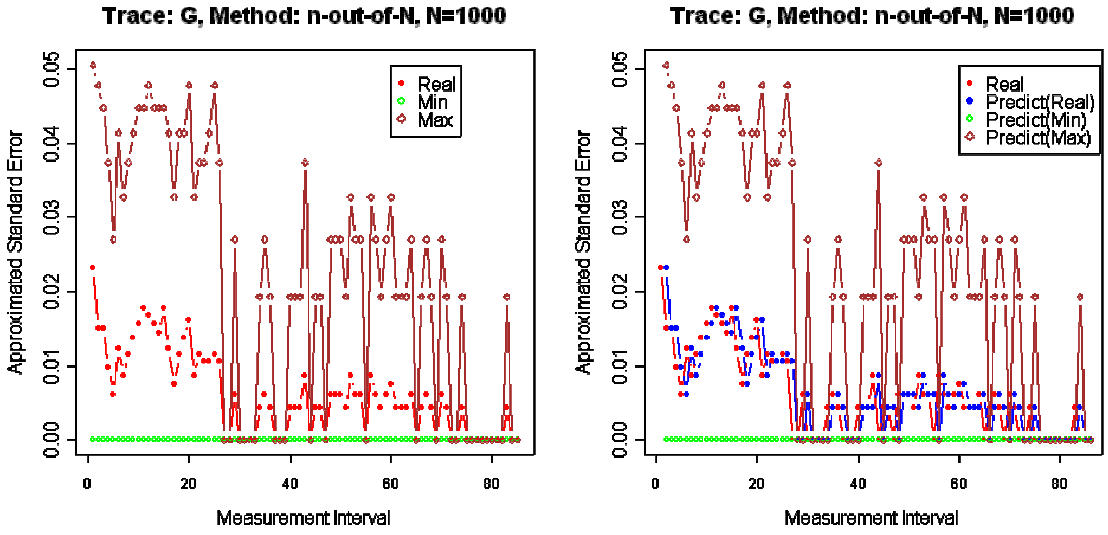


Figure 7-26: Predicted Absolute Standard Error for all Measurement Intervals in Trace G (N=1000)

No real improvement of the prediction can be observed if smaller measurement intervals are used. For trace G a slightly better prediction can be observed. The absolute difference between the predicted standard error (predicted from the real proportion of the previous

measurement interval) and the theoretical expected standard error is on average a little bit smaller (0.0003 instead of 0.0005) if smaller measurement intervals are used. But for trace B the difference is even larger on average (0.006 instead of 0.002) than for the experiments with small measurement intervals. So for the investigated traces the prediction could not be improved by using smaller measurement intervals.

In trace G there are several measurement intervals where the proportion is 0 (e.g., MI 27, 28, 30, 31, 32, 33, etc.). In those measurement intervals all packets conform to the SLA and therefore it is impossible to select a violator. That means for every run the estimated proportion is $\hat{P} = 0$. Therefore also all approximated standard errors from all runs are zero. One can see in Figure 7-26 that as expected the minimum and the maximum approximated standard error are zero in these intervals.

7.6.2.2 Confidence Limits

As in section 7.6.1.2 above, it is now investigated what effect the usage of the predicted standard error has on the calculation of the confidence limits and therefore on the confidence level. Again 100,000 sample runs are performed with a sample fraction of 5%. For each run a different estimate can occur and with this different CI limits. Then it is checked in how many of the 100,000 runs the real proportion lies within the calculated confidence limits. From theory a confidence level of 95% is expected. The four prediction methods described in 7.6.2.1 are compared.

The following figures show the percentage of sample runs for which the real proportion was within the calculated confidence limits (i.e., the CI contains the real value). The colors in the graphs mean the following:

- Red line: statistically expected percentage of runs with real proportion in CI limits (from theory)
- Orange dots ("*Real*"): percentage of runs with real proportion in CI limits, if the limits are calculated with the real standard error (calculated with the real proportion)
- Green triangles (point-up) ("*Approximated*"): percentage of runs with real proportion in CI limits, if the standard error (and with this the CI limits) is calculated from the estimated proportion in the current measurement interval
- Blue triangles (point-down) ("*Predict*"): percentage of runs with real proportion in CI limits, if the standard error is calculated from the real proportion in the previous measurement interval
- Brown circles ("*PredictApprox*"): percentage of runs with real proportion in CI limits, if the standard error is calculated from the estimated proportion in the previous measurement interval

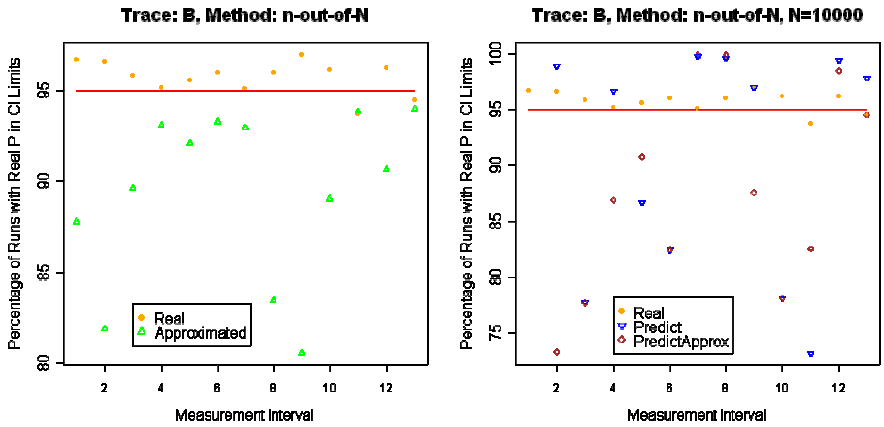


Figure 7-27: Percentage of Runs where Real Proportion is within CI Limits (per MI), Different Prediction Methods (Trace B, N=10,000, Sample Fraction 5%, Threshold 168051.2µs)

Figure 7-27 shows the results for trace B and small measurement intervals. If the real standard error is used to calculate the CI limits, all values are close to the expected 95%. For most measurement intervals one can even observe a higher percentage of runs, where the real proportion lies in the CI limits. That mean if one could calculate the CI limits with the real proportion, one would get the theoretically expected confidence level. If the estimated standard error (from the samples in the actual measurement interval) is used instead, the real values lie in less runs within the CI. One gets a lower percentage than statistically expected for all measurement intervals. That means one gets a smaller confidence level than expected. The smallest percentage is 80.57 %, the largest is 93.99 %.

The left diagram shows the results if the standard error is calculated with the real or estimated proportion from the previous interval. Values between 73.22 and 99.77 can be observed for the calculation with the real proportion from the previous MI and values between 73.31 and 99.90 for the calculation with the estimated proportion from the previous MI.

Trace, MIlength	Prediction Method	Min	1st Quartile	Median	Mean	3rd Quartile	Max
B, N=10k	Real	93.69	95.13	95.96	95.71	96.20	96.96
B, N=10k	Approx	80.57	87.79	90.68	89.41	93.10	93.99
B, N=10k	Predict	73.22	81.34	96.80	90.60	99.00	99.77
B, N=10k	PredictApprox	73.31	81.34	87.22	87.67	95.46	99.90

Table 7-10: Comparison of Methods for Trace B

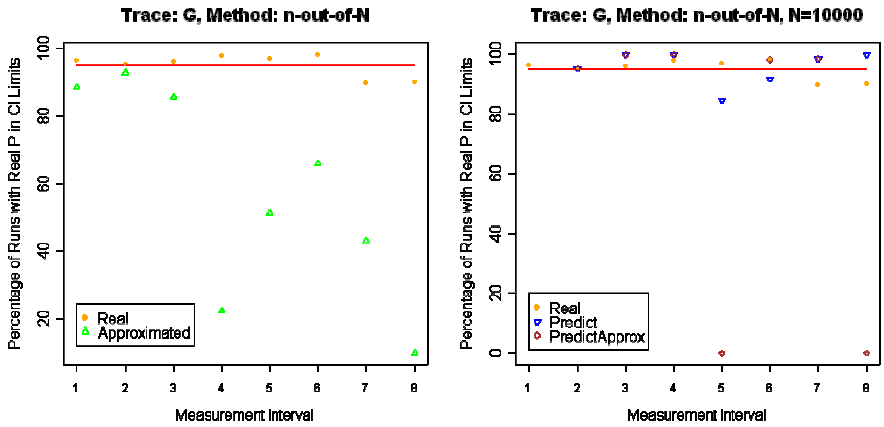


Figure 7-28: Percentage of Runs where Real Proportion is within CI Limits (per MI), Different Prediction Methods (Trace G, N=10,000, Sample Fraction 5%, Threshold 16001µs)

Figure 7-28 shows the results for trace G and small measurement intervals. Again the calculation with the real proportion is close to the theoretically expected 95%. If the approximated values from the estimated proportion of the current MI are used, one gets smaller values between 9.88 % and 92.80 %. For some measurement intervals they are much smaller than the results for trace B. For the measurement intervals 4 and 8 one gets the smallest values. These were the measurement intervals with the smallest violator proportion (see Table 7-9).

The results from the prediction with the real proportion from the previous interval look a little bit better than the prediction for trace B, but for the prediction from the estimated proportion from the previous interval one even gets a percentage of 0 for some measurement intervals (5 and 8).

Trace, MIlength	Prediction Method	Min	1st Quartile	Median	Mean	3rd Quartile	Max
G, N=10k	Real	89.82	93.92	96.22	95.06	97.20	98.16
G, N=10k	Approx	9.881	37.860	58.560	57.420	86.320	92.800
G, N=10k	Predict	84.54	93.43	98.46	95.60	99.74	99.88
G, N=10k	PredictApprox	0.00	47.59	98.16	70.20	99.10	99.88

Table 7-11: Comparison of Methods for Trace G

7.6.3 Conclusion

In order to provide an interval estimate, one calculates confidence limits from the estimate and the standard error. The standard error depends on the unknown real violator proportion in the measurement interval. Since the value is not known in reality different methods to approximate or predict this value from current and previous samples are compared.

For this many sample runs are performed on the traces. Then it is calculated in how many of the runs the real value lies within the confidence interval, which is calculated with different prediction methods. This percentage is compared to the theoretically expected confidence level.

The experiments showed that the percentage of runs is very close to the expected confidence level if the standard error is calculated with the real violator proportion. If an approximation from samples of the current MI is used, one gets a smaller percentage for all measurement intervals of the investigated traces. But if the proportion itself is not too small (only a few packets), one still gets a very high number of runs for with the CI contains the real value. Furthermore, the experiments were performed with a sample fraction of 5% and it was shown that the percentage increases for larger sample fractions. So the approximation of the standard error with the samples from the current MI introduces only a small modification of the confidence limit and is applicable if the proportions and the sample fraction are not too low.

No trends could be observed for the violator proportions of subsequent MIs. Therefore, a prediction of the proportions from previous MIs is not very promising. If one predicts the current proportion by the proportion of the previous MI, one gets a smaller percentage of runs where the real value lies within the CI for some MIs and larger percentages for other MIs. If the estimated proportion from the previous interval is used for the prediction, one gets for most MIs a percentage below the expected confidence level and the values can get very small. Experiments with smaller MIs showed similar poor prediction results. The error can be immense if one relies on predicted values. The investigated prediction methods are not suitable to provide a basis for adaptive sampling.

7.7 Stratification

Intelligently deployed stratification techniques can lead to an improvement of the estimation accuracy without increasing the sample size or to a reduction of the sample size without reducing the accuracy. But such gain can only be achieved if one finds a suitable stratification variable (see 3.4.1).

The survey variable here is the conformance of the packet to the SLA. This conformance status is derived from the one-way delay per packet by classifying packets in accordance to a pre-defined delay threshold. One candidate for a stratification variable is the packet size. The packet size can be obtained by looking into the packet header whereas the calculation of the delay requires not only the processing but also the transfer of per packet information. Therefore the packet size can be easier obtained than the delay. Furthermore, it is possible that there is a correlation between the packet size and the packet delay (see e.g., [RFC889], [ChMC03]).

It is first checked whether there is a correlation between the packet size and the packet delay. If traces with such correlation are found, it is further investigated, whether also the conformance status of the packets is correlated to the packet size. For traces where the

conformance status has a significant correlation to the packet sizes it is possible to achieve a gain by using stratification techniques based on the packet size.

7.7.1 Correlation between Packet Size and Delay

In the traces B, D and F all packets have the same size (except some outliers in trace D). It is obvious that one cannot use the packet size as stratification variable for those traces. Therefore only the traces A, C, E, G, H, I and J are investigated. For the investigations the whole traces with all measurement intervals are considered.

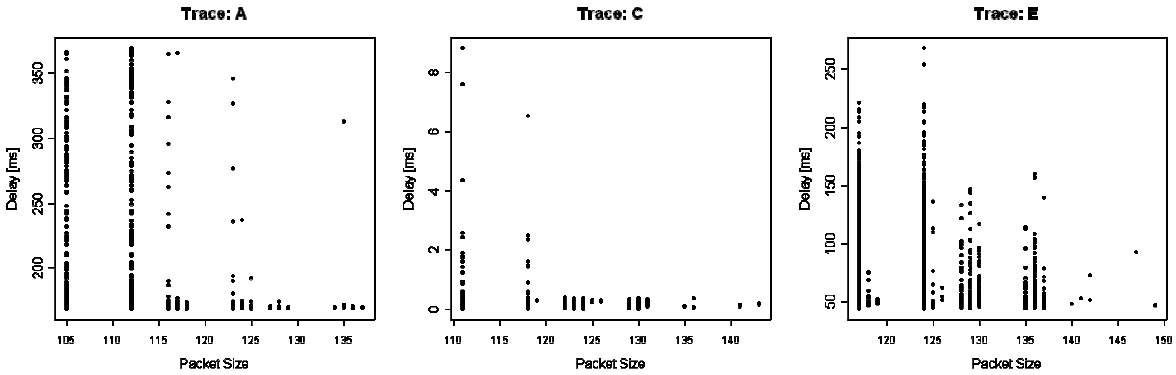


Figure 7-29: Correlation of Packet Sizes and Delays in Gaming Traces A, C, and E

Figure 7-29 shows the scatterplots for packet size and delay for the gaming traces. As already observed in 7.4.2 the gaming traces consist only of small packets. Furthermore, some packet sizes are not present in the traces. From the graphs one can see no correlation between packet sizes and packet delay.

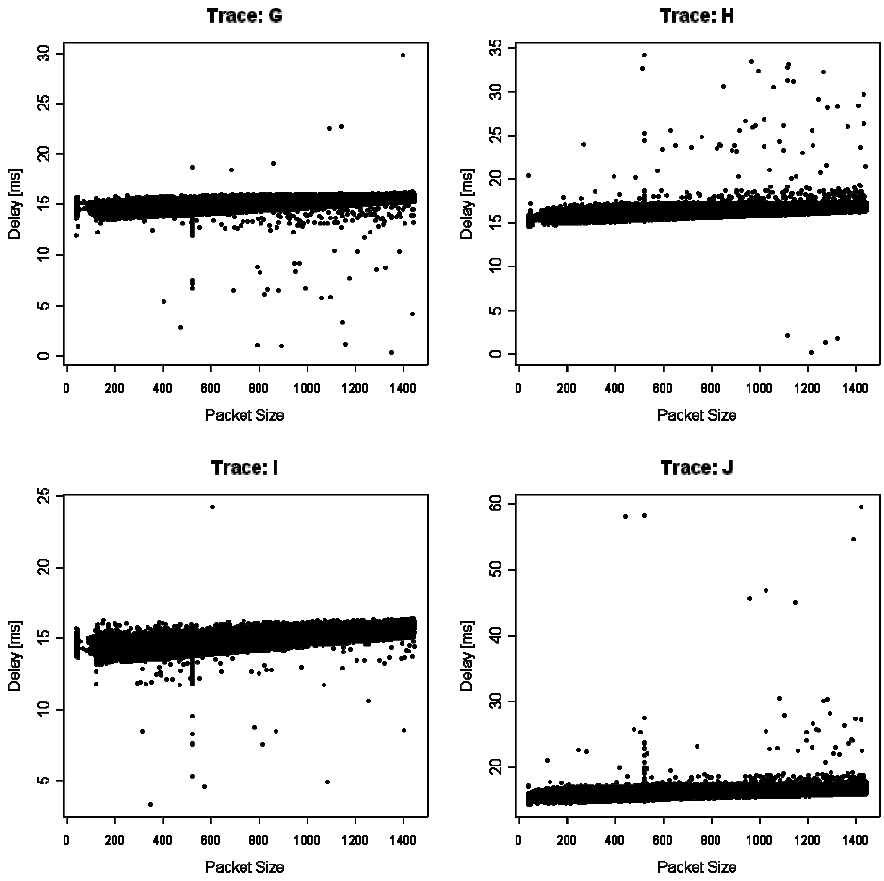


Figure 7-30: Correlation of Packet Sizes and Delays in Video Traces G, H, I, and J

The video traces contain much more different packet sizes than the gaming traces (see also boxplots in section 7.4.2). The scatterplots show that larger packets observe a little bit higher delays. Table 7-12 shows the correlation coefficients for the traces:

Trace	Correlation
A	0.01715979
C	-0.01619174
E	0.02324112
G	0.7300941
H	0.541415
I	0.7186552
J	0.510603

Table 7-12: Correlation between Packet Sizes and Delay

There is nearly no correlation between packet sizes and delay in the gaming traces. For the gaming traces the packet size is no suitable stratification variable. For the video traces one gets much higher correlation coefficients. Therefore the packet size could be a potential stratification variable for the video traces. But for the SLA validation only the conformance status of the packet (derived form the delay) is considered and not the delay itself. Therefore

one needs to check whether also the conformance of packets to the SLA is correlated with the packet size (e.g., if large packets are more often violators than small packets).

7.7.2 Correlation between Packet Size and Conformance Status

The correlation of the conformance status of the packets with the packet size is only investigated for the video traces. In the gaming traces there was no correlation between delay and packet size. Therefore no correlation is expected between the conformance status and the packet size for the gaming traces. The packets of the video trace are classified into conformant and non-conformant packets. For this different thresholds are used. The threshold values per trace are taken from the percentiles of the delay distribution of the whole trace[19] (Table 7-13).

Trace	1st	5th	25th	50th	75th	95th	99th
G	13.943	14.054	14.709	15.158	15.540	15.857	15.959
H	15.22333	15.59900	16.13000	16.37500	16.65700	17.04100	17.15200
I	13.6750	13.8770	14.2530	14.6555	15.1060	15.8330	16.1030
J	15.101	15.468	16.013	16.301	16.637	17.069	17.270

Table 7-13: Percentiles of Delay Distributions (in ms)

Packets are classified in accordance to the thresholds in Table 7-13. Then the correlation coefficient is calculated for the conformance status and the packet size. The results are shown in Table 7-14.

Trace	Threshold from Percentile						
	1st	5th	25th	50th	75th	95th	99th
G	0.1621781	0.3111584	0.6747087	0.6593934	0.4885649	0.2695545	0.1294455
H	0.24525898	0.38705582	0.54636042	0.47062389	0.33293510	0.19022838	0.07462975
I	0.1081310	0.1776571	0.3576254	0.5462066	0.7003467	0.4408718	0.2100660
J	0.18773917	0.34127524	0.51926096	0.44708201	0.35912324	0.17405866	0.07831511

Table 7-14: Correlation between Conformance and Packet Size for Different Thresholds

The highest overall correlation occurs in trace I if the classification into conformant and non-conformant packets is done with a delay threshold of 15.106 ms. If one uses this threshold one get the following violator proportions and correlation coefficients per measurement interval.

[19] Here the delay distribution of the whole trace is taken into account. Therefore the percentiles differ from those shown in Table 7-6, which are the percentiles of the first measurement interval.

MI	1	2	3	4	5	6
Proportion	0.3700	0.3025	0.3088	0.2520	0.1904	0.1525
Correlation	0.6925636	0.7094542	0.7162472	0.7479161	0.7625209	0.6956605

Table 7-15: Violator Proportion and Correlation Coefficients for Different Measurement Intervals

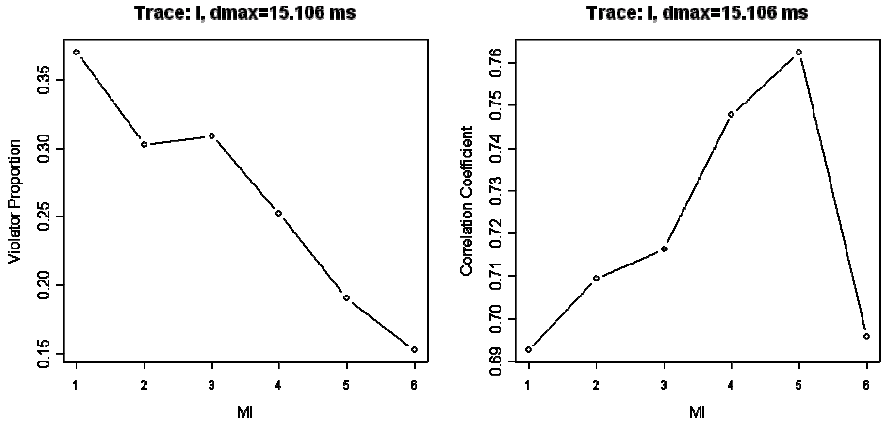


Figure 7-31: Violator Proportions and Correlation Coefficients per Measurement Interval for Trace I with Threshold d_{max}=15.106 ms

7.7.3 Standard Error for Stratified Sampling

In order to check whether a gain can be achieved with stratification, multiple stratification experiments were performed with the most promising trace. The expected gain depends on the correlation between the survey variable (here SLA conformance) and the stratification variable (here the packet size). The highest correlation was observed in trace I when using a delay threshold of $d_{max}=15.106$ *ms*. Therefore the experiments were performed with trace I classified with that threshold.

The experiments were performed on all measurement intervals. In all experiments 2 strata are defined and a proportional allocation was used. With two strata one only need to set one intermediate boundary. Stratum 1 contains all packets with sizes between the minimum packet size and the intermediate boundary. Stratum 2 contains all packets with sizes between the intermediate boundary and the maximum packet size. In the experiments the intermediate boundary was varied, to investigate its influence on the stratification gain.

For each boundary setting, the number N_l of elements in each stratum l and the proportion of violators P_l in each stratum was calculated. From these values the expected standard error with formula (6.40) from section 6.7 is calculated. Then the standard error from stratified sampling is compared with the expected standard error for random sampling in order to see what gain can be achieved with stratification.

If the boundary is set to 0 bytes (minimum theoretical packet size) or to 1500 bytes (maximum theoretical packet size), all packets belong to the same stratum. It is therefore expected to get the same standard error as for random sampling for those cases. Table 7-16 shows the results of the experiments. It shows the absolute standard errors for stratifications with different boundaries for the six measurement intervals of trace I.

Boundary [Bytes]	**MI 1**	**MI 2**	**MI 3**	**MI 4**	**MI 5**	**MI 6**
None (Random)	0.02159167	0.02054233	0.02066120	0.01941628	0.01755835	0.01607755
0	0.02159167	0.02054233	0.02066120	0.01941628	0.01755835	0.01607755
100	0.02149783	0.02049539	0.02060153	0.01936966	0.01753325	0.01605550
200	0.02145122	0.02042350	0.02055014	0.01932862	0.01750439	0.01603814
300	0.02137856	0.02038879	0.02047703	0.01928570	0.01746635	0.01601916
400	0.02124195	0.02024874	0.02030502	0.01915677	0.01738086	0.01596159
500	0.02092987	0.01993323	0.01999786	0.01889550	0.01719381	0.01584028
600	0.01680170	0.01611811	0.01594065	0.01510162	0.01396060	0.01368834
700	0.01606307	0.01528271	0.01510266	0.01396841	0.01292742	0.01295769
800	0.01615998	0.01495749	0.01495994	0.01314743	0.01194018	0.01212833
900	0.01614223	0.01514345	0.01501659	0.01325761	0.01146964	0.01147361
1000	0.01636111	0.01530896	0.01524344	0.01360834	0.01170741	0.01140582
1100	0.01673763	0.01552081	0.01534551	0.01376600	0.01182740	0.01168438
1200	0.01776930	0.01628705	0.01633180	0.01426094	0.01233052	0.01226451
1300	0.01905286	0.01775764	0.01797616	0.01578671	0.01295111	0.01296486
1400	0.02073262	0.01961064	0.01979959	0.01817972	0.01576822	0.01448954
1500	0.02159167	0.02054233	0.02066120	0.01941628	0.01755835	0.01607755

Table 7-16: Absolute Standard Error for Different Boundaries and Different Measurement Intervals (Trace I, d_{max}=15.106 ms)

The difference between the standard error for random sampling and the standard error for stratified sampling is illustrated in Figure 7-32. The left diagram in Figure 7-32 shows the absolute standard error for different measurement intervals. The red curve shows the standard error for random sampling as a reference. The other curves show the standard error for stratified sampling with 2 strata and the boundary at 200, 600, 1000, and 1400 bytes. One can see that for all measurement intervals and for all boundary settings the standard error for stratified sampling lies below the standard error for random sampling. That means one indeed achieves a higher accuracy with stratified sampling with the same sample size. As expected the standard error for boundary 0 bytes and boundary 1500 bytes (both not shown in diagram) are equal to the standard error for random sampling.

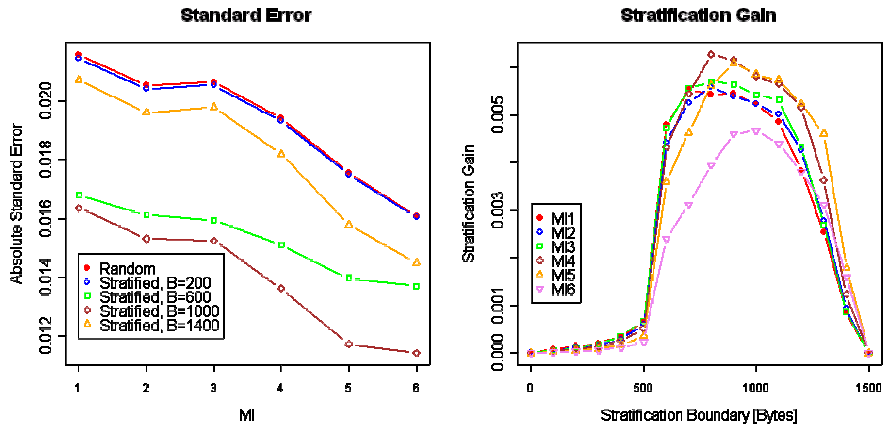


Figure 7-32: Standard Error and Stratification Gain for Different Boundaries and Measurement Intervals (Trace I, d_{max}=15.106 ms)

The right diagram of Figure 7-32 shows the stratification gain Δ_{SE}, calculated as the difference between the standard error from random sampling and the standard error from stratified sampling. Here the x-axis shows the boundary setting and the different curves are for different measurement intervals. Please note that here the colors differentiate between different measurement intervals whereas in the left diagram the colors are used to differentiate between different boundary settings.

It is now checked what the gain means for the estimation accuracy. For this the limits of the confidence interval achieved with stratified sampling are compared with those from random sampling. As an example the first measurement interval is used and a stratification boundary of 1000 bytes. For a confidence level of 95% the critical value $z_c=1.96$ is used, for a confidence level of 99% the critical value is $z_c=2.58$.

Method	**Confidence Level**	**StdErr**	**ε**	**CI Limits**
Random	95%	0.0216	0.0423	$\hat{P} \pm 0.0423$
Stratified	95%	0.0164	0,0321	$\hat{P} \pm 0.0321$
Random	99%	0.0216	0,0557	$\hat{P} \pm 0.0557$
Stratified	99%	0.0164	0,0422	$\hat{P} \pm 0.0422$

Table 7-17: Comparison of CI Limits

One can see that for stratified sampling the CI limits are smaller. In the first measurement interval the real violator proportion is 0.37. The estimates for stratified sampling lie in 99% of all cases between 0.37-0.0422=0.3278 and 0.37+0.0422=0.4122. For random sampling they lie between 0.37-0.0557=0.3143 and 0.37+0.0557=0.4257. So, one can achieve a small accuracy improvement by using stratified sampling.

7.7.4 Conclusion

In this chapter it was investigated whether the use of stratified sampling can improve the estimation accuracy, without the need to increase the sample size. The packets size was used as stratification variable. In the investigated gaming traces the correlation between packet size and delay was too small, to expect any benefit from stratification. For the video traces stronger correlations were observed. The standard error and stratification gain was calculated for the most promising trace. Experiments were performed for all measurement intervals in the trace and with different stratification boundaries. With some boundary settings an accuracy improvement could be achieved.

Nevertheless, even for the most correlated traces, the accuracy improvement was rather marginal in the experiments. Furthermore, stratification requires additional classification effort and the setting of optimal boundaries requires some a-priori knowledge about the distribution of the survey variable.

Therefore it is recommended to use stratification only if a very strong correlation (larger than the correlation coefficient of 0.7, observed in our experiments) is expected between the survey variable and the stratification variable and if the setting of stratification boundaries can be based on some prior knowledge about the distribution of the survey variable.

8 Conclusion and Suggestions for Future Work

In this work sampling schemes for non-intrusive measurements were investigated with regard to achievable accuracy, resource consumption and the required traffic information. Usage-based accounting and SLA validation were used as target applications. First a taxonomy for packet selection methods was introduced and related work in this area was evaluated. Most promising schemes were selected. For those schemes mathematical models were developed for the assessment of the achievable accuracy under consideration of the available traffic information and resource consumption. It was shown when and how the estimation accuracy depends on different traffic characteristics. Experiments with real traffic traces were performed to evaluate results from modeling. The experiments were also used to explore how the accuracy evolves if initial assumptions about traffic characteristics, which were required for the modeling, are violated. Furthermore, it was investigated to what degree a prediction of the accuracy is possible if only available information before and after the selection process is taken into account. A special focus was set on the investigation of stratified schemes for both target scenarios.

8.1 Sampling for Usage-based Accounting

For the usage-based accounting scenario systematic, n-out-of-N sampling and a count-based stratified method (1-inK sampling) were investigated for the estimation of the volume transmitted per flow.

It was shown by mathematical modeling how the achievable accuracy depends on various traffic characteristics. For the modeling it was necessary to distinguish different cases with regard to available traffic information and to postulate some initial assumptions. Empirical investigations showed that within the given limitations the results from the sampling simulation are very close to those values expected from theoretical modeling. Nevertheless, from the trace investigation it became clear that not all assumptions hold true in reality. With additional experiments it was shown that the modeling also remains applicable with more relaxed assumptions.

For the investigated traces the stratified scheme (1-in-K sampling) performed slightly better than n-out-of-N sampling. Especially for large flows often a higher accuracy could be achieved with 1-in-K sampling. Since the effort for realizing this stratified scheme is comparable to the realization of n-out-of-N sampling, an accuracy gain can be achieved with nearly no additional effort. Systematic sampling performs quite well in terms of number of flows that get a higher accuracy. Nevertheless, the systematic sampling showed very varying and unpredictable results for the investigated flows. Potential correlations can never be excluded. Therefore the results for systematic sampling have to be considered as trace-specific and cannot be generalized for arbitrary traces.

In the experiments the most critical traffic characteristic for the accuracy assessment was the flow size. All sampling schemes showed a much higher accuracy for large flows than for small flows. A comparison to common accuracy requirements showed that a reasonable accuracy can be achieved only for flows that comprise of a larger percentage (>10%) of the packets in the measurement interval. Modification of classification rules or the measurement interval lengths help to achieve higher accuracies but there are limits due to the overall flow duration.

Nevertheless, separate accounting of flows that comprise only of few packets is probably not desired anyway. If many small flows occur, it is much better to apply more coarse classification rules or to summarize all small flows into one traffic class. Adaptive flow sampling methods can be applied for the selection of flows with regard to the expected estimation accuracy. The neglecting of flows with insufficient accuracy and the application of post aggregation methods is currently investigated in the follow-up of the VEGAS project (VEGAS-II). For this, the developed n-out-of-N model serves as basis for the assessment of the expected accuracy that can then be used as input for the flow selection process.

For all schemes the estimation accuracy depends on traffic characteristics that are unknown in reality. Using worst case traffic characteristics as input to the model in most cases is too inaccurate for an assessment of the estimation accuracy. A better option is to estimate those traffic characteristics from sampled values. Experiments were used to investigate how close these approximated accuracy values are to the real accuracy. For large flows and large sample fraction the approximations get very close to the real accuracy. For very small flows or small sample fractions the approximation of the estimation accuracy was unacceptably inaccurate. This increases the incentive to adapt classification rules or to combine packet sampling with flow sampling in order to concentrate on the large flows as explained above.

For the practical deployment of sampling it is important that sufficient information is stored not only for the calculation of the estimate but also for the assessment of the estimation accuracy. In this work it was shown that it is essential to store at least the sum and the squaresum of the bytes of the sampled packets to provide an accuracy statement. Cisco NetFlow currently only stores the sum of the packet sizes. Therefore storing the squaresum was one strong recommendation given to Cisco engineers.

Storing the squaresum does not only simplify the calculation of the estimation accuracy for sampled measurements. It is also of high value if the whole population is measured because it allows the calculation of the variance of the packet sizes per flow. With this it reveals a huge more insight into network traffic in general and is an excellent example how a little well-selected piece of information can provide a further dimension of knowledge. Cisco Systems has now decided to provide an additional counter for the squaresum in their routers in future. In addition to this, the squaresum was also added as an information element (octetDeltaSumOfSquares) to the IPFIX standard [QuBM05].

8.2 Sampling for SLA Validation

Up to now many providers and users use active measurement to validate SLA guarantees. In this work the use of passive measurements for SLA validation is motivated and the use of sampling methods is proposed to reduce the amount of measurement data.

In this work the validation of SLA guarantees is modeled as estimation of the proportion of packets that violate the contract. The method is better suited for SLA validation than the estimation of percentiles, which provides an insight about the network situation but doesn't reveal the amount of non-conforming packets.

In this work systematic sampling, random n-out-of-N sampling, two variants of probabilistic sampling and a content-based stratified sampling that uses the packet size as stratification variable were investigated.

With theoretical modeling and experiments on real traffic traces it was shown how the estimation accuracy for different schemes depends on traffic characteristics. The accuracy achieved in experiments was close to the expected accuracy from the models. The expected estimation accuracy for all schemes increases, the closer the number of non-conformant packets is to the number of conformant packets.

If only a small percentage of packets do not conform to the SLA guarantees, probabilistic sampling showed quite similar results as n-out-of-N sampling. For larger proportion of non-conformant packets n-out-of-N provides slightly better results.

For probabilistic sampling a higher accuracy can be achieved if the real sample size n_R is available for the extrapolation instead of the target sample size n_T. That can be simply realized by providing an additional counter. The systematic sampling performed quite well for the investigated traces. Nevertheless, the accuracy of systematic sampling highly depends on the sequence at which non-conformant packets occur in the trace. Since periodicities are likely if congestion leads to non-conformance, the results cannot be generalized for arbitrary traces.

Different methods were investigated to approximate and predict the estimation accuracy.

This was done by comparing the achieved level of confidence with the theoretical expected. If the standard error is calculated with the real traffic characteristics, the results are very close to the theoretically expected values.

If the standard error is calculated by using estimated traffic characteristics from the samples of the current measurement interval the values differ only slightly from the expected accuracy. Especially, if the number of non-conformant packets in the trace is not too small, results get close to those that were calculated with the real traffic characteristics.

The use of predicted traffic characteristics from previous measurement intervals showed much worse results. Since no trend in the relevant traffic characteristics for subsequent measurement intervals was observed, the prediction did not lead to satisfactory results.

8.3 Suggestions for Future Work

Parts of the work on volume estimation are continued in the project VEGAS-II. Based on VEGAS-I results, the project now focuses on a further improvement of the estimation accuracy. One approach is the combination of packet sampling with a controlled flow selection based on the expected estimation accuracy per flow and a flexible post-aggregation. With this, flows for which the expected accuracy is insufficient for given accuracy requirements can be separated and further aggregated until a sufficient accuracy level can be achieved.

Another idea is the assignment of an individual measurement interval per flow. With the availability of link counters, the measurement interval can be reduced for each flow to the identified flow duration. This technique reduces the population to the relevant interval and with this optimizes the estimation accuracy per flow. Furthermore, it fits well in the flow cache based concept of NetFlow, works with all NetFlow flow termination criteria and can be realized with few additional operations.

Regarding SLA measurements more complex techniques considered in the PSAMP group are of interest. Hash-based techniques provide a valuable method to synchronize selection processes at multiple observation points. The investigation of hash-based methods was not explicitly addressed in this work. It was expected that investigations started by others will shortly provide exploitable results. Nevertheless, up to now research results are still not sufficient to motivate a general use of hash-based methods. The emulation of a random selection by performing a deterministic function on the packet content is a difficult task. The emulation quality, especially the degree of independence of the selection from packet attributes, was empirically checked for specific traces and hash functions in [MoND05]. The tests were intended to provide recommendations for the PSAMP group for the use of hash-based selection methods in [ZsMD05]. Although it was decided to include some recommendations into the PSAMP standard, the evaluation results in [MoND05] are very specific to the investigated functions, attributes and traces. They cannot be generalized to arbitrary traffic traces, other packet attributes or different hash-functions. Furthermore, no tests have been performed with IPv6 traffic. IPv6 packets seem to have less variation than IPv4 packets in their header fields, e.g., due to the address structure and the lack of an identification field. Furthermore, IPv6 allows extension headers, which makes it more difficult to identify the start of the payload. Therefore some further difficulties are expected when applying a hash-based selection to IPv6 traffic. In addition to this, it is quite likely that IPv6 traffic profiles change significantly in future when IPv6 is used by a broader community. That means IPv6 traffic traces measured today are far from being representative for future IPv6 traffic. The investigation of hash-based methods for IPv4 and IPv6 is still a quite new research area and an interesting field for potential future work.

Further applications with growing importance that require measurement support originate from the need to protect the network against attacks. Sophisticated denial of service attacks, propagation of Internet worms and other viruses cause serious damage at global scale and severely endanger the operation of the Internet. Attackers permanently adapt to defense strategies, making it necessary to react with more sophisticated detection methods. The ultimate goal here is to cope with zero-day events, where new unknown attack patterns occur for the first time. Depending on the attack type, a suitable detection requires traffic measurements at different time scales, per flow information and often also per packet information. It may require correlation of data from multiple measurement points, efficient post processing and the incorporation of historic data from previous measurements.

The CERT® Coordination Center [CERT/CC] collected a wish list of network information that would be useful to detect incidents [Long04]. They clearly state the need for data reduction techniques like sampling and aggregation and observe standardization efforts like IPFIX and PSAMP. Due to the diversity of attacks, a variety of metrics may be of interest for the detection. A smart deployment of packet and flow selection methods is extremely valuable to reduce the amount of data and can also dynamically direct the data selection to the relevant events. Bandwidth-oriented or claim-and-hold attacks can also introduce a threat originated from the measurement system itself. Measurement of attack data can lead to an overwhelming amount of measurement data that additionally overloads the network. The deployment of adaptive data selection methods can prevent this additional building up of traffic load.

For security applications also the coupling of measurement operations with AAA functions and the exchange of measurement data between domains provides an extremely valuable combination that helps to establish an efficient attack detection and defense. The coordination and control of data selection and aggregation techniques in cooperation with AAA functions and other domains are valuable building blocks for the defense against network attacks. Approaches for AAA controlled measurement operations can be found in [RFC3334]. Ideas for coupling IPFIX and AAA functions are introduced in [ZsBB05].

A further interesting field is the provisioning of a higher flexibility to data selection and aggregation techniques. Customer demands and the advent of IPFIX have triggered the development of many further features in Cisco NetFlow, allowing a much higher flexibility for measurement configuration. If, in addition to this, a fast re-configuration of such functions is possible further applications with the combination of selection and aggregation techniques are imaginable. Such techniques could be used to support adaptive selection schemes. Furthermore, re-configuration can be used to zoom into suspicious traffic on demand or look from different viewpoints at the data e.g., if an anomaly is detected. Measurement functions could adapt themselves with regard to the current measurement needs, currently available resources or external factors like threat levels, etc.

A self-configuration of measurement functions e.g., with regard to the amount of suspicious traffic can be made possible by allowing measurement functions in one network node to

communicate their observations to neighbor nodes. Such functions provide important components towards autonomic communication in IP networks.

9 References

[6QM] IPv6 QoS Measurements (6QM) Project, http://www.6qm.org

[AdMa00] Andrew K. Adams, Matthew Mathis. *A System for Flexible Network Performance Measurement*, Proceedings of INET 2000 The Internet Global Summit, Yokohama, Japan, July 2000. Available: http://www.isoc.org/inet2000/

[AmCa89] Paul D. Amer, Lillian N. Cassel. *Management of Sampled Real-Time Network Measurements*. 14th Conference on Local Computer Networks, Minneapolis, pages 62-68, IEEE, October 1989.

[ATTRD] AT&T Research, http://www.research.att.com/facilities/fp/

[BMWG] Benchmarking Working Group, IETF Working Group, http://www.ietf.org/html.charters/bmwg-charter.html

[BoBC04] Herbert Bos, Willem de Bruijn, Mihai Cristea, Trung Nguyen, Georgios Portokalidis. *FFPF: Fairly Fast Packet Filters*. Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI'04), San Francisco, USA, December 2004.

[BrCl03] Stewart Bryant, Benoit Claise, *IPFIX Default Transport*. Presentation at IPFIX meeting, Proceedings of 58th IETF , November 2003.

[BRO] BRO Intrusion Detection System, http://www.bro-ids.org/

[Brow99] Nevil Brownlee. *NeTraMet & NeMaC: Reference Manual.* Version 4.3, Information Technology Systems & Services, The University of Auckland, Auckland, New Zealand, June 1999.

[CAIDA] Cooperative Association for Internet Data Analysis (CAIDA), http://www.caida.org/

[CERT/CC] CERT Coordination Center, http://www.cert.org

[ChMC03] Baek-Young Choi, Sue Moon, R. Cruz, Zhi-Li Zhang, Christophe Diot. *Practical Delay Measurements for ISPs.* SPRINT ATL research report RR03-ATL-051910, May 2003.

[ChMZ04] Baek-Young Choi, Sue Moon, Zhi-Li Zhang, Papagiannaki Konstantina, Christophe Diot. *Analysis of Point-To-Point Packet Delay in an Operational Network*. Proceedings of IEEE INFOCOM 2004, Hong Kong, March 2004.

[ChPZ02a] Baek-Young Choi, Jaesung Park, Zhi-Li Zhang. *Adaptive Random Sampling for Load Change Detection*. ACM SIGMETRICS 2002, Marina Del Rey, CA, USA, June 2002.

[ChPZ02b] Baek-Young Choi, Jaesung Park, Zhi-Li Zhang. *Adaptive Packet Sampling for Flow Volume Measurement*. Student Poster Session, ACM SIGCOMM 2002, August 2002.

[ChPZ02c] Baek-Young Choi, Jaesung Park, Zhi-Li Zhang. *Adaptive Packet Sampling for Flow Volume Measurement*. ACM SIGCOMM Computer Communication Review, Vol. 32, No. 3, July 2002.

[Claf02] K.C. Claffy. *Internet Measurement: Myths about Internet Data*. Presented at North American Network Operators' Group (NANOG 24), February 2002, Available: www.caida.org/outreach/presentations/Myths2002/

[Claf03] K.C. Claffy. *Priorities and Challenges in Internet Measurement, Simulation, and Analysis*. Presented at the NSF PI meeting, Reston, VA, January 9-10, 2003.

[Clai05] B. Claise (Editor). *IPFIX Protocol Specification*. Internet Draft <draft-ietf-ipfix-protocol-17.txt>, work in progress, July 2005.

[ClDG00] J. Cleary, S. Donnelly, I. Graham, A. McGregor, M. Pearson. *Design Principles for Accurate Passive Measurement.* Passive and Active Measurement Workshop (PAM 2000), Hamilton, New Zealand, April 2000.

[ClPB93] K.C. Claffy, George C Polyzos, Hans-Werner Braun. *Application of Sampling Methodologies to Network Traffic Characterization*. Proceedings of ACM SIGCOMM 1993, San Francisco, CA, USA, September 1993.

[ClPB95] K. C. Claffy, Hans-Werner Braun, George C. Polyzos. *A Parameterizable Methodology for Internet Traffic Flow Profiling*. IEEE Journal on Selected Areas in Communications, vol. 13, no. 8, pages 1481-1494, October 1995.

[Coch72] William G. Cochran. *Stichprobenverfahren*. Walter de Gruyter &Co, Berlin, New York, 1972.

[CoGi98] I. Cozzani, S. Giordano. *Traffic Sampling Methods for end-to-end QoS Evaluation in Large Heterogeneous Networks.* Computer Networks and ISDN Systems, 30 (16-18), September 1998.

[CoMN04] J. Coppens, E.P. Markatos, J. Novotny, M. Polychronakis, V. Smotlacha, S. Ubik. *SCAMPI - A Scaleable Monitoring Platform for the Internet.* Proceedings of the 2nd International Workshop on Inter-Domain Performance and Simulation (IPS 2004), Budapest, Hungary, 22-23 March 2004.

[CoSu03] Cemal Coemert, Juraj Sučík. *Non-Intrusive QoS Measurements in Multimedia Environments.* 2nd International Conference on Emerging Telecommunications Technologies and Applications (ICETA 2003), Kosice, Slovakia, September 11-13, 2003.

[CoSV04] Jan Coppens, Stijn De Smet, Steven Van den Berghe, Filip De Turck, Piet Demeester; *Performance Evaluation of a Probabilistic Packet Filter Optimization Algorithm for High-speed Network Monitoring*. Proceedings of 7th IEEE International Conference on High Speed Networks and Multimedia Communications (HSNMC 2004), Toulouse, France, 30 June - 2 July 2004.

[CrHM03] Jon Crowcroft, Steven Hand, Richard Mortier, Timothy Roscoe, Andrew Warfield. *QoS's downfall: at the bottom, or not at all!* Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS (RIPQoS): What have we learned, why do we care?, Karlsruhe, Germany, pages 109 – 114, 2003, ACM Press New York, NY, USA, ISBN:1-58113-748-6.

[CSTB01] Committee on Research Horizons in Networking, Computer Science and Telecommunications Board (CSTB), Division on Engineering and Physical Sciences, National Research Council. *Looking Over the Fence at Networks: A Neighbor's View of Networking Research (2001)*. National Academy Press, Washington, D.C., 2001.

[DAG] The DAG project. http://dag.cs.waikato.ac.nz

[DCFa] Arbeitsweise und Genauigkeitsvergleich DCF77 und GPS Zeitempfänger, http://www.hopf-time.com/de/dcf-gps.htm

[DCFb] Das DCF77 Funksignal, http://www.dcf77.com/

[Deri03a] Luca Deri. *Passively Monitoring Networks at Gigabit Speeds using Commodity Hardware and Open Source Software*. Proceedings of Passive and Active Measurement Workshop (PAM 2003), La Jolla, CA, USA, April 6-8, 2003.

[Deri03b] Luca Deri. *nProbe: an Open Source NetFlow Probe for Gigabit Networks*. Proceedings of Terena TNC 2003 Conference, Zagreb, Croatia, May 2003.

[DiPP04] Miguel Ángel Díaz, Jordi Palet, Guido Pohl, Emile Stephan, Lidia Yamamoto, David Diep. *First Year Public Trial and Evaluation Report*. 6QM Project Deliverable D4.4, July 2004. Available: http://www.6qm.org/deliverables.php

[DrCh98] J. Drobisz, K. J. Christensen. *Adaptive Sampling Methods to determine network traffic statisitics including the Hurst parameter*. Proceedings of IEEE Annual Conference on Local Computer Networks, pages 238-247, 1998.

[Duff04] Nick Duffield. *Sampling for Passive Internet Measurement: A Review.* Statistical Science,Vol. 19, No. 3, 472–498, 2004.

[Duff05] Nick Duffield (Editor). *A Framework for Passive Packet Measurement*. Internet Draft <draft-ietf-psamp-framework-10.txt>, work in progress, January 2005.

[DuGG02] Nick Duffield, A. Gerber, Matthias Grossglauser. *Trajectory Engine: A Backend for Trajectory Sampling*. IEEE Network Operations and Management Symposium 2002, Florence, Italy, April 15-19, 2002.

[DuGr00] Nick Duffield, Matthias Grossglauser. *Trajectory Sampling for Direct Traffic Observation*. Proceedings of ACM SIGCOMM 2000, Computer Communications Review, vol. 30, no. 4, pages 271-282, October 2000.

[DuGr01] Nick Duffield, Matthias Grossglauser. *Trajectory Sampling for Direct Traffic Observation*. IEEE/ACM Transactions on Networking, vol 9, no. 3, pages 280-292, June 2001.

[DuGr03] Nick Duffield, Matthias Grossglauser. *Trajectory Sampling*. White Paper, 2003. Available: http://www.research.att.com/~duffield/pubs/ts_white_paper.pdf

[DuGr04] Nick Duffield, Matthias Grossglauser. *Trajectory Sampling with Unreliable Reporting.* IEEE INFOCOM 2004, Hong Kong, March 7-11, 2004.

[DuLT01] Nick Duffield, Carsten Lund, Mikkel Thorup. *Charging from Sampled Network Usage*. ACM Internet Measurement Workshop IMW 2001, San Francisco, USA, November 1-2, 2001.

[DULT02] Nick Duffield, Carsten Lund, Mikkel Thorup. *Properties and Prediction of Flow Statistics from Sampled Packet Streams*. ACM SIGCOMM Internet Measurement Workshop IMW 2002, Marseille, France, November 6-8, 2002.

[DuLT03] Nick Duffield, Carsten Lund, Mikkel Thorup. *Estimating Flow Distributions from Sampled Flow Statistics*. ACM SIGCOMM 2003, Karlsruhe, Germany, August 25-29, 2003.

[DuLT04] Nick Duffield, Carsten Lund, Mikkel Thorup. *Flow Sampling Under Hard Resource Constraints*. ACM SIGMETRICS 2004, New York, USA, June 12-16, 2004.

[DuLT05a] Nick Duffield, Carsten Lund, Mikkel Thorup. *Learn more, sample less: control of volume and variance in network measurement*, IEEE Transactions in Information Theory, vol. 51, no. 5, pages 1756-1775, 2005.

[DuLu03] Nick Duffield, Carsten Lund. *Predicting resource usage and estimation accuracy in an IP flow measurement collection infrastructure*. Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement (IMC), Miami Beach, FL, USA , pages 179-191, 2003, ISBN:1-58113-773-7.

[ENDACE] Endace Measurement Systems. http://www.endace.com

[ErMS01] F. Ergun, S. Mittra, S. Sahinalp, J. Sharp, R. Sinha. *A Dynamic Lookup Scheme for Bursty Access Patterns.* In Proceedings of IEEE INFOCOM 2001, Anchorage, AK, USA, 2001.

[EsKM04] Cristian Estan, Ken Keys, David Moore, George Varghese. *Building a Better NetFlow*. ACM SIGCOMM 2004, Portland, OR, USA, September 2004.

[EsVa02a] Cristian Estan, George Varghese. *New Directions in Traffic Measurement and Accounting*. UCSD technical report CS2002-0699, February 2002.

[EsVa02b] Cristian Estan, George Varghese. *New Directions in Traffic Measurement and Accounting.* ACM SIGCOMM 2002, Pittsburgh, PA, USA, August 2002.

[EsVa03] Cristian Estan, George Varghese. *New Directions in Traffic Measurement and Accounting: Focusing on the Elephants, Ignoring the Mice*. ACM Transactions on Computer Systems, August 2003.

[FaPe99] Wenjia Fang, Larry Peterson. *Inter-AS Traffic Patterns and their Implications*. Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 1999), no. 1, pages 1859-1868, December 1999.

[FeGL00] Anja Feldmann, Albert Greenberg, Carsten Lund, Nick Reingold, Jennifer Rexford, Fred True. *Deriving Traffic Demands for Operational IP Networks: Methodology and Experience*. ACM SIGCOMM Computer Communication Review, vol. 30, issue 4, pages 257-270, October 2000, ACM Press, New York, NY, USA, ISSN:0146-4833.

[FeGL01] Anja Feldmann, Albert Greenberg, Carsten Lund, Nick Reingold, Jennifer Rexford, Fred True. *Deriving traffic demands for operational IP networks: methodology and experience*. IEEE/ACM Transactions on Networking, 9(3), pages 265-280, 2001.

[FeMu00] Anja Feldmann, S. Muthukrishnan. *Tradeoffs for Packet Classification*. In Proceedings of INFOCOM 2000, vol. 3, pages 1193-1202, IEEE, March 2000.

[Fisz63] Marek Fisz. *Probability Theory and Mathematical Statistics*. Third Edition, Robert E. Krieger, Publishing Company Inc., Malabar, Florida 1963

[FlPa01] Sally Floyd, Vern Paxson. *Difficulties in Simulating the Internet*. IEEE/ACM Transactions in Networking, vol. 9, no. 4, pages 392-403, 2001.

[FrDL01] C. Fraleigh, C. Diot, B. Lyles, S. Moon, P. Owezarski, D. Papagiannaki, F. Tobagi. *Design and Deployment of a Passive Monitoring Infrastructure*. Proceedings of Passive and Active Measurement Workshop (PAM 2001), Amsterdam, The Netherlands, April 2001.

[FrMe94] V. Frost and B. Melamed. *Traffic Modeling for Telecommunications Networks. IEEE Communications Magazine*, 32(3), pages 70-80, March, 1994.

[G.711] ITU-T Recommendation G.711, Pulse Code Modulation (PCM) Of Voice Frequencies, 1993.

[GrDM98] Ian D. Graham, Stephen F. Donnelly, Stele Martin, Jed Martens, John G. Cleary. *Nonintrusive and Accurate Measurement of Unidirectional Delay and Delay Variation on the Internet*. INET 1998, Geneva, Switzerland, 21-24 July, 1998

[GSL] Mark Galassi, Jim Davies, James Theiler, Brian Gough, Gerard Jungman, Michael Booth, Fabrice Rossi. *GNU Scientific Library: Reference Manual*. Network Theory Ltd., 2nd edition, February 1, 2003, ISBN: 0954161734

[HeBh03] Tristan Henderson, Saleem Bhatti. *Networked games - a QoS-sensitive application for QoS-insensitive users?* Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS (RIPQoS): What have we learned, why do we care?, Karlsruhe, Germany, 2003, ACM Press New York, NY, USA, ISBN:1-58113-748-6.

[HoVe03] Nicolas Hohn, Darryl Veitch. *Inverting Sampled Traffic*. ACM SIGCOMM Internet Measurement Conference (IMC 2003), Miami Beach, Florida, USA, October 2003.

[Hust03] Geoff Huston. *Measuring IP Network Performance.* The Internet Protocol Journal, March 2003.

[IaDG01] G. Iannaccone, C. Diot, I. Graham, N. McKeown. *Monitoring Very High Speed Links*. ACM SIGCOMM Internet Measurement Workshop (IMW 2001), San Francisco, USA, November 1-2, 2001.

[ICIR] ICSI Center for Internet Research. http://www.icir.org/

[ICSI] International Computer Science Institute. http://www.icsi.berkeley.edu/

[IDMAPS] Internet Distance Maps. http://idmaps.eecs.umich.edu/

[IETF] Internet Engineering Task Force. www.ietf.org

[IMRG] Internet Measurement Research Group. http://www.irtf.org/charters/imrg.html

[INFRA] Internet Measurement Infrastructure. http://www.caida.org/analysis/performance/measinfra/

[Internet-2] Internet-2. http://www.internet2.edu/

[IPFIX] IP Flow Information Export Working Group, IETF Working Group. http://www.ietf.org/html.charters/ipfix-charter.html

[IPMON] IPMon Project. http://ipmon.sprint.com/

[IPPM] IP Performance Metrics Working Group, IETF Working Group. http://www.ietf.org/html.charters/ippm-charter.html

[IRTF] Internet Research Task Force. http://www.irtf.org

[ITU-T] International Telecommunications Union, Telecommunication Standardization Sector (ITU-T). http://www.itu.int/home/

[Jaco97] Van Jacobson. *pathchar - A Tool to Infer Characteristics of Internet Paths*. Talk at Mathematical Science Research Institute (MSRI), April 1997, Available: ftp://ftp.ee.lbl.gov/pathchar/msri-talk.ps.gz

[JaLM01] V. Jacobson, C. Leres, S. McCanne. *tcpdump manual page*. Lawrence Berkeley National Laboratory, University of California, Berkeley, CA, USA, 2001.

[JaRo86] R. Jain, S. Routhier. *Packet trains - measurements and a new model for computer network traffic*. IEEE Journal of Selected Areas in Communications, Vol. SAC-4, No. 6, Pages 986–995, September 1986.

[JePP92] Jonathan Jedwab, Peter Phaal, Bob Pinna. *Traffic Estimation for the Largest Sources on a Network, Using Packet Sampling with Limited Storage*. HP technical report, Management, Mathematics and Security Department, HP Laboratories, Bristol, March 1992. Available: http://www.hpl.hp.com/techreports/92/HPL-92-35.html

[KoLM04] M. Kodialam, T. V. Lakshman, Shantidev Mohanty. *Runs bAsed Traffic Estimator (RATE): A Simple, Memory Efficient Scheme for Per-Flow Rate Estimation*. IEEE INFOCOM 2004, Hong Kong, March 7-11, 2004.

[KuXW04] Abhishek Kumar, Jun Xu, Jia Wang, Oliver Spatscheck, Li Li. *Space-Code Bloom Filter for Efficient Per-Flow Traffic Measurement*. IEEE INFOCOM 2004, Hong Kong, March 7-11, 2004.

[LeTW94] W. Leland, M. Taqqu, W. Willinger, D. Wilson. *On the Self-Similar Nature of Ethernet Traffic(Extended Version)*. IEEE ACM Transactions on Networking, vol.2, no.1, pages 1-15, February 1994.

[Long04] Tom Longstaff. *Wish List*. Proceedings of FloCon 2004, Crystal City, July 2004. Available: http://www.cert.org/flocon/2004/proceedings.html

[LuMc02] M. J. Luckie, A. J. McGregor. *IPMP: IP Measurement Protocol*. Proceedings of Passive and Active Measurement Workshop (PAM2002), Fort Collins, Colorado, pages 168-176, March 2002.

[MaBP03] Carlos Macian, Lars Burgstahler, Wolfgang Payer, Sascha Junghans, Christian Hauser, Juergen Jaehnert. *Beyond Technology: The Missing Pieces for QoS Success.* Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS (RIPQoS): What have we learned, why do we care?, Karlsruhe, Germany, 2003, ACM Press New York, NY, USA, ISBN:1-58113-748-6.

[MaVa94] J. K. MacKie-Mason, H. Varian. *Pricing the Internet*. Second International Conference on Telecommunications Systems, Modelling and Analysis, Nashville, Tennesse, USA, March 1994.

[McGr02] Tony McGregor. *Quality in measurement: beyond the deployment barrier*. Proceedings of the Symposium on Applications and the Internet (SAINT) Workshop 2002, pages 66-73
28 Jan.-1 Feb. 2002.

[MoCR04] A. Morton, L. Ciavattone, G. Ramachandran, S. Shalunov, J. Perser. *Packet Reordering Metric for IPPM*. Internet Draft <draft-ietf-ippm-reordering-08.txt>, Work in Progress, December 2004.

[Moli03] Maurizio Molina. *A Scalable and Efficient Methodology for Flow Monitoring in the Internet*, International Teletraffic Congress (ITC-18), Berlin, September 2003.

[MOME] Coordinating Action on Monitoring and Measurement (MOME). http://www.ist-mome.org

[MoND05] Maurizio Molina, Saverio Niccolini, Nick Duffield. *A Comparative Experimental Study of Hash Functions Applied to Packet Sampling*. International Teletraffic Congress (ITC-19), Beijing, August 2005.

[MoUK04] Tatsuya Mori, Masato Uchida, Ryoichi Kawahara, Jianping Pan, Shigeki Goto. *Identifying Elephant Flows Through Periodically Sampled Packets*. ACM Internet Measurement Conference (IMC 2004), October 25–27, 2004, Taormina, Sicily, Italy, 2004.

[MuCl01] Margaret Murray, K.C. Claffy. *Measuring the Immeasurable: Global Internet Measurement Infrastructure*. Proceedings of Passive and Active Measurements Workshop (PAM2001), 2001.

[NetFlow] *NetFlow Services and Applications*. Cisco White Paper. Available: http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm

[NFperf02] *NetFlow Performance Analysis*. Cisco White Paper. Available: http://www.cisco.com/warp/public/cc/pd/iosw/prodlit/ntfo_wp.pdf

[NFperf04] *CISCO IOS Sampled NetFlow*., Internet technologies Division, November 2004. Available: http://www.cisco.com/warp/public/732/Tech/nmp/docs/sampled_netflow.pdf

[NgCB04] Trung Nguyeny, Mihai Cristeay, Willem de Bruijn, Herbert Bos. *Scalable network monitors for high-speed links: a bottom-up approach*. Proceedings of 2004 IEEE International Workshop on IP Operations & Management (IPOM'04), Beijing, China, 11-13 October 2004.

[NGI] Next Generation Internet. http://www.ngi.gov/

[NiMD04] Saverio Niccolini, Maurizio Molina, Nick Duffield. *Evaluation of Hash Functions for Passive Inter-Domain Measurements*. 6QM Measurement Workshop, Berlin, Germany, December 14, 2004. Available: http://www.6qm.org/workshop/slides/niccolini-6QM-PSAMP-hashcomparison.pdf

[NiMT04] Saverio Niccolini, Maurizio Molina, Sandra Tartarelli, F. Raspall. *Design and implementation of a One Way Delay passive measurement system*. Proceedings of IEEE/IFIP Network Operations and Management Symposium (NOMS 2004), 2004.

[NLANR] The National Laboratory for Applied Network Research (NLANR). http://www.nlanr.net/

[NLAPa] NLANR Packets, NLANR Newsletter. http://www.nlanr.net/NLANRPackets/

[NLTraces] NLANR Traces. http://moat.nlanr.net/ Traces/Traces/

[NTP] The Network Time Protocol, http://www.ntp.org

[OsHe02] Arne Øslebø, Jon Kåre Hellan (Editors). *D0.2: Measurement-based Application Requirements*. SCAMPI Deliverable, October 8, 2002. Available: http://www.ist-scampi.org/publications/deliverables/#wp0

[Osle02] Arne Øslebø (Editor). *D0.1: Description and Analysis of the State-of-the-Art*. SCAMPI Deliverable D01. SCAMPI Deliverable, September 17, 2002. Available: http://www.ist-scampi.org/publications/deliverables/#wp0

[PaAM00] Vern Paxson, Andrew Adams, Matt Mathis. *Experiences with NIMI*. Proceedings of Passive and Active Measurement Workshop (PAM2000), Hamilton, New Zealand, April 3-4, 2000.

[PaFl95] Vern Paxson, Sally Floyd. *Wide area Traffic: The Failure of Poisson Modelling*. IEEE/ACM Transactions on Networking 3, pages 226-244, 1995.

[PaMF02] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, C. Diot. *Measurement and analysis of single-hop delay on an IP backbone network*. Proceedings of IEEE INFOCOM, San Francisco, April 2002.

[PaSF02] Jia-Yu Pan, Srinivasan Seshan, Christos Faloutsos. *FastCARS: Fast, Correlation-Aware Sampling for Network Data Mining.* Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 2002), Taipei, Taiwan, November 17-21, 2002.

[Paxs04] Vern Paxson. *Strategies for Sound Internet Measurement*. Proceedings of ACM Internet Measurement Conference (IMC 2004), Taormina, Sicily, Italy, October 25-27, 2004.

[Paxs99] Vern Paxson. *Bro: A System for Detecting Network Intruders in Real-Time*. Computer Networks, 31(23-24), pages 2435-2463, December 14, 1999.

[Peuh01] Markus Peuhkuri. *A Method to Compress and Anonymize Packet Traces*. Proceedings of ACM Internet Measurement Workshop (IMW 2001), San Francisco, California, USA, November 1-2, 2001.

[PSAMP] Packet Sampling Working Group, IETF Working Group. http://www.ietf.org/html.charters/psamp-charter.html

[QuBM05] J. Quittek, S. Bryant, J. Meyer. *Information Model for IP Flow Information Export*. Internet Draft <draft-ietf-ipfix-info-09>, work in progress, July 2005.

[RandNF] *Random Sampled NetFlow.* Cisco Technical Documentation. Available: http://www.cisco.com/en/US/products/sw/iosswrel/ps5207/products_feature_guide09186a00801a7618.html

[RFC1112] Steve Deering. *Host Extensions for IP Multicasting*. RFC 1112, August 1989.

[RFC1305] Dave Mills. *Network Time Protocol (Version 3) Specification, Implementation*. RFC 1305, March 1992.

[RFC1633] R. Braden, D. Clark, S. Shenker. *Integrated Services in the Internet Architecture: an Overview*, RFC 1633, June 1994.

[RFC1930] J. Hawkinson, T. Bates. *Guidelines for creation, selection, and registration of an Autonomous System (AS)*. RFC 1930, March 1996.

[RFC2123] Nevil Brownlee. *Traffic Flow Measurement: Experiences with NeTraMet*. RFC 2123, March 1997.

[RFC2205] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin. *Resource Reservation Protocol (RSVP) Version 1 Functional Specification*, RFC 2205, September 1997.

[RFC2330] V. Paxson, G. Almes, J. Mahdavi, M. Mathis. *Framework for IP Performance Metrics*, RFC 2330, May 1998.

[RFC2402] S. Kent, R. Atkinson. *IP Authentication Header*. RFC 2402, November 1998.

[RFC2406] S. Kent, R. Atkinson. *IP Encapsulating Security Payload (ESP)*. RFC 2406, November 1998.

[RFC2460] Steve Deering, R. Hinden. *Internet Protocol, Version 6 (IPv6) Specification*. RFC 2460, December 1998.

[RFC2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss. *An Architecture for Differentiated Services*. RFC 2475, December 1998.

[RFC2638] K. Nichols, V. Jacobson, L. Zhang. *A Two-bit Differentiated Services Architecture for the Internet*. RFC 2638, July 1999.

[RFC2678] J. Mahdavi, V. Paxson. *IPPM Metrics for Measuring Connectivity*, RFC 2678, September 1999.

[RFC2679] G. Almes, S. Kalidindi, M. Zekauskas. *A One-way Delay Metric for IPPM*. RFC 2679, September 1999.

[RFC2680] G. Almes, S. Kalidindi, M. Zekauskas. *A One-way Packet Loss Metric for IPPM*. RFC 2680, September 1999.

[RFC2681] G. Almes, S. Kalidindi, M. Zekauskas. *A Round-trip Delay Metric for IPPM*, RFC 2681, September 1999.

[RFC2721] Nevil Brownlee. *RTFM: Applicability Statement*. RFC 2721, October 1999.

[RFC2722] Nevil Brownlee, C. Mills, G. Ruth. *Traffic Flow Measurement: Architecture*. RFC 2722, October 1999.

[RFC2960] R. Stewart, Q. Xie, K. Morneault, C. Sharp,H. Schwarzbauer, T. Taylor, L. Rytina, M. Kalla, L. Zhang, V. Paxson. *Stream Control Transmission Protocol*. RFC 2960, October 2000.

[RFC3176] P. Phaal, S. Panchen, N. McKee. *InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks*. RFC 3176, September 2001.

[RFC3260] D. Grossman. *New Terminology and Clarifications for Diffserv*. RFC 3260, April 2002.

[RFC3334] Tanja Zseby, Sebastian Zander, Georg Carle. *Policy-Based Accounting*. RFC 3334, October 2002.

[RFC3357] R. Koodli, Ravikanth. *One-way Loss Pattern Sample Metrics*. RFC 3357, August 2002.

[RFC3393] C. Demichelis, P. Chimento. *IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)*, RFC 3393, November 2002.

[RFC3487] Henning Schulzrinne. *Requirements for Resource Priority Mechanisms for the Session Initiation Protocol (SIP).* RFC 3487, February 2003.

[RFC3550] Henning Schulzrinne, S. Casner, R. Frederick, V. Jacobson. *RTP: A Transport Protocol for Real-Time Applications*. RFC 3550, July 2003.

[RFC3689] K. Carlberg, R. Atkinson. *General System Requirements for Emergency Telecommunications Service.* RFC 3689, February 2004.

[RFC3690] K. Carlberg, R. Atkinson. *IP Telephony Requirements for Emergency Telecommunications Service.* RFC 3690, February 2004.

[RFC3758] R. Stewart, M. Ramalho, Q. Xie, M. Tuexen, P. Conrad. *Stream Control Transmission Protocol (SCTP) Partial Reliability Extension*. RFC 3758, May 2004.

[RFC3763] S. Shalunov, B. Teitelbaum. *One-way Active Measurement Protocol (OWAMP) Requirements*. RFC 3763, April 2004.

[RFC3917] Jürgen Quittek, Tanja Zseby, Benoit Claise, Sebastian Zander. *Requirements for IP Flow Information Export (IPFIX)*. RFC 3917, October 2004.

[RFC3954] Benoit Claise (Editor). *Cisco Systems NetFlow Services Export Version 9*. RFC 3954, October 2004.

[RFC768] J. Postel. *User Datagram Protocol*. RFC 768, August 1980.

[RFC791] J. Postel. *Internet Protocol*. RFC 791, September 1981.

[RFC792] J. Postel. *Internet Control Message Protocol*. RFC 792, September 1981.

[RFC793] J. Postel. *Transmission Control Protocol*. RFC 793, September 1981.

[RFC889] D. Mills. *Internet Delay Experiments*, RFC 889, December 1983.

[Rinn97] Horst Rinne. *Taschenbuch der Statistik*. Verlag Harri Deutsch, Frankfurt am Main, 1997.

[RIPE] RIPE. http://www.ripe.net/

[RMON] Remote Monitoring Working Group (RMON), IETF Working Group. http://www.ietf.org/html.charters/rmonmib-charter.html

[RyCB01] B. Ryu, D. Cheney, H. Braun. *Internet Flow Characterization: Adaptive Timeout Strategy and Statistical Modeling*. Proceedings of Passive and Active Measurement Workshop (PAM 2001), Amsterdam, The Netherlands, April 23-24, 2001.

[SaAl00] Kamil Sarac, Kevin C. Almeroth. *Supporting Multicast Deployment Efforts: A Survey of Tools for Multicast Monitoring (2001)*. Journal of High Speed Networking-Special Issue on Management of Multimedia Networking, 2001.

[SaBC05] G. Sadasivan, N. Brownlee, B. Claise, J. Quittek. *Architecture for IP Flow Information Export*. Internet Draft < draft-ietf-ipfix-architecture-08.txt>, work in progress, March 2005.

[Sada01] Ganesh Sadasivan. *Scalability Analysis Of Netflow Export using TCP*. Presentation at IPFIX meeting, Proceedings of 52nd IETF , Salt Lake City, December, 2001.

[SCAMPI] A Scaleable Monitoring Platform for the Internet (SCAMPI) Project. http://www.ist-scampi.org

[Schm01] Carsten Schmoll. *Dynamically Configurable Network Meter for Accounting in IP-based Networks*. Diploma Thesis, Technical University Berlin, December 2001.

[Schw75] H. Schwarz. *Stichprobenverfahren*, Oldenbourg Verlag GmbH, 1975.

[SFLOW] sFlow. http://www.sflow.org/

[ShTe03] Stanislav Shalunov, Benjamin Teitelbaum *Quality of Service and Denial of Service.* Proceedings of the ACM SIGCOMM workshop on Revisiting IP QoS (RIPQoS): What have we learned, why do we care?, Karlsruhe, Germany, 2003, ACM Press New York, NY, USA, ISBN:1-58113-748-6.

[ShTK04] Stanislav Shalunov, Benjamin Teitelbaum, Anatoly Karp, Jeff W. Boote, Matthew J. Zekauskas. *A One-way Active Measurement Protocol (OWAMP)*. Internet Draft <draft-ietf-ippm-owdp-14.txt>, work in progress, December 2004.

[Srin01] V. Srinivasan. *A Packet Classification and Filter Management System*. Proceedings of IEEE INFOCOM 2001, April 2001.

[STARTAP] The Science, Technology, And Research Transit Access Point (STARTAP). http://www.startap.net/

[Tann03] Andrew S. Tannenbaum. Computer Networks.Fourth Edition, Pearson Education International Inc., Prentice Hall PTR, 2003, ISBN 0-13-038488-7.

[TEWG] http://www.ietf.org/html.charters/tewg-charter.html

[TOOLS] http://www.ip-measurement.org/tools/avail.html

[VEGAS-SW] Tanja Zseby, Alexander Widmann. *VEGAS Traffic Analysis and Sampling Simulation Software*. VEGAS Deliverable, April 2005.

[WAND] WAND group. http://wand.cs.waikato.ac.nz/

[WaSV01] Priyank Warkhede, Subhash Suri, Georg Varghese. *Fast Packet Classification for Two-Dimensional Conflict-Free Filters.* Proceedings of IEEE INFOCOM 2001, Anchorage, AK, USA, 2001.

[WeOw75] E. S. Wentzel, L. A. Owtscharow. *Aufgabensammlung zur Wahrscheinlichkeitsrechnung*. Berlin, Akademieverlag, 1975.

[WITS] Waikato Internet Traffic Storage. http://wand.cs.waikato.ac.nz/wand/wits/

[Woo00] Thomas Y.C. Woo. *A Modular Approach to Packet Classification: Algorithms and Results.* Proceedings of IEEE INFOCOM 2000, March 2000.

[XuFA01] Jun Xu, Jinliang Fan, Mostafa Ammar, Sue B. Moon. *On the Design and Performance of Prefix-Preserving IP Traffic Trace Anonymization*. Proceedings of ACM Internet Measurement Workshop (IMW 2001), San Francisco, California, USA, November 1-2, 2001.

[Yurc04] William Yurcik. Sharing Intelligence is our Best Defense: Incentives That Work versus Disincentives That Can Be Solved. Proceedings of FloCon 2004, Crystal City, July 2004. Available: http://www.cert.org/flocon/2004/proceedings.html

[ZsBB05] Tanja Zseby, Elisa Boschi, Nevil Brownlee, Benoit Claise. IPFIX Applicability, Internet Draft <draft-ietf-ipfix-as-06.txt>, work in progress, July 2005.

[Zseb02] Tanja Zseby. *Deployment of Sampling Methods for SLA Validation with Non-Intrusive Measurements*. Proceedings of Passive and Active Measurement Workshop (PAM 2002), Fort Collins, CO, USA, March 25-26, 2002

[Zseb03] Tanja Zseby. *Stratification Strategies for Sampling-based Non-intrusive Measurements of One-way Delay.* Proceedings of Passive and Active Measurement Workshop (PAM 2003), La Jolla, CA, USA, April 6-8, 2003.

[Zseb04] Tanja Zseby. *Comparison of Sampling Methods for Non-Intrusive SLA Validation*. 2nd Workshop on End-to-End Monitoring Techniques and Services (E2EMON), October 3, 2004.

[ZsMD05] Tanja Zseby, Maurizio Molina, Nick Duffield, Saverio Niccolini, Fredric Raspall. *Sampling and Filtering Techniques for IP Packet Selection.* Internet Draft <draft-ietf-psamp-sample-tech-07.txt>, work in progress, July 2005.

[ZsZa03] Tanja Zseby, Sebastian Zander. *Statistical SLA Validation Based on Sampling for Highly Interactive Applications*. Fraunhofer FOKUS Technical Report TR-2003-1101, November 2003.

[ZsZa04] Tanja Zseby, Sebastian Zander. *Sampling Algorithms for Validating Service Level Agreements*. CAIA Technical Report 040706A. Centre for Advanced Internet Architectures, Swinburne University of Technology, July 2004.

[ZsZC01] Tanja Zseby, Sebastian Zander, Georg Carle. *Evaluation of Building Blocks for Passive One-way-delay Measurements.* Proceedings of Passive and Active Measurement Workshop (PAM 2001), Amsterdam, The Netherlands, April 23-24, 2001.

A Terminology

In this section the terminology is defined that is used throughout this document. Terminology is mainly adopted from the definitions of the PSAMP and IPFIX group in [Duff05], [RFC3917] and [ZsMD05]. Some additional terms were defined and some descriptions where modified slightly. It is clearly marked in which document the terms are defined. Terms without a reference are own definitions.

Non-intrusive Measurements: Non-intrusive measurements are based only on existing traffic in the network. They are called non-intrusive, because they do not inject any test traffic. They are also called passive measurements.

Intrusive Measurements: Intrusive measurements are based on the sending of test traffic. They are also called active measurements.

Measurement Interval: Interval for which a measurement statement should be made. It can be defined as time interval or in number of packets. The measurement interval length can differ with regard to the measurement purpose (e.g., smaller intervals for quality validation than for accounting) and data rates.

Observation Point: An observation point is a location in the network where a packet stream is observed. Examples are a line to which a probe is attached, a shared medium, such as an Ethernet-based LAN, a single port of a router, or set of interfaces (physical or logical) of a router or an embedded measurement subsystem within an interface. [RFC3917]

Metering Process: The metering process generates flow records. Input to the process are packet headers observed at an observation point and packet treatment at the observation point, for example the selected output interface. The metering process consists of a set of functions that includes packet header capturing, timestamping, sampling, classifying, and maintaining flow records. The maintenance of flow records may include creating new records, updating existing ones, computing flow statistics, deriving further flow properties, detecting flow expiration, passing flow records to the exporting process, and deleting flow records. [RFC3917]

IP Traffic Flow: A flow is defined as a set of IP packets passing an observation point in the network during a certain time interval. All packets belonging to a particular flow have a set of common properties. Each property is defined as the result of applying a function to the values of:

- one or more packet header field (e.g., destination IP address), transport header field (e.g., destination port number), or application header field (e.g., RTP header fields [RFC3550])
- one or more characteristics of the packet itself (e.g., number of MPLS labels, etc.)
- one or more of fields derived from packet treatment (e.g., next hop IP address, the output interface, etc...)

A packet is defined to belong to a flow if it completely satisfies all the defined properties of the flow. This definition covers the range from a flow containing all packets observed at a network interface to a flow consisting of just a single packet between two applications with a specific sequence number. [RFC3917]

Flow Record: A flow record contains information about a specific flow that was metered at an observation point. A flow record contains measured properties of the flow (e.g., the total number of bytes of all packets of the flow) and usually also characteristic properties of the flow (e.g., source IP address). [RFC3917]

Exporting Process: The exporting process sends flow records to one or more collecting processes. The flow records are generated by one or more metering processes. [RFC3917]

Collecting Process: The collecting process receives flow records from one or more exporting processes. The collecting process might store received flow records or further process them, but these actions are out of the scope of this document. [RFC3917].

Packet Stream: a sequence of packets, each of which was observed at the observation point. Note that when packets are sampled from a stream, the selected packets usually do not have common properties by which they can be distinguished from packets that have not been selected. Therefore we define here the term stream instead of flow, which is defined as set of packets with common properties [RFC3917]. [ZsMD05]

Observed Packet Stream: The Observed Packet Stream is the set of all packets observed at the Observation Point. [ZsMD05]

Selection Process: A selection process takes the observed packet stream as its input and selects a subset of that stream as its output. [ZsMD05]

Selector: A Selector defines the action of a selection process on a single packet of its input. If selected, the packet becomes an element of the output packet stream. The Selector can make use of the following information in determining whether a packet is selected:

- the packet's content;
- information derived from the packet's treatment at the observation point;
- any selection state that may be maintained by the selection process. [ZsMD05]

Filtering: A filter is a selector that selects a packet deterministically based on the packet content, or its treatment, or functions of these occurring in the selection state. Examples include match Filtering, and Hash-based Selection. [ZsMD05]

Sampling: A selector that is not a filter is called a sampling operation. This reflects the intuitive notion that if the selection of a packet cannot be determined from its content alone, there must be some type of Sampling taking place. [ZsMD05]

Content-independent Sampling: A Sampling operation that does not use packet content (or quantities derived from it) as the basis for selection is called a content-independent Sampling operation. Examples include systematic Sampling, and uniform pseudorandom Sampling

driven by a pseudorandom number whose generation is independent of packet content. Note that in content-independent Sampling it is not necessary to access the packet content in order to make the selection decision. [ZsMD05]

Content-dependent Sampling: A Sampling operation where selection is dependent on packet content is called a Content-dependent Sampling operation. Examples include pseudorandom selection according to a probability that depends on the contents of a packet field. Note that this is not a filter, because the selection is not deterministic. [ZsMD05]

Packet Content: The packet content denotes the union of the packet header (which includes link layer, network layer and other encapsulation headers) and the packet payload. Note that packets selected from a stream, e.g., by Sampling, do not necessarily possess a property by which they can be distinguished from packets that have not been selected. For this reason the term "stream" is favored over "flow", which is defined as set of packets with common properties [RFC3917]. [ZsMD05]

Configured Selection Fraction: The Configured Selection Fraction is the ratio of the number of packets selected by a Selector from an input Population, to the Population Size, as based on the configured selection parameters. [ZsMD05] Please note that in this work the term **target sampling fraction** is used instead.

Attained Selection Fraction: The Attained Selection Fraction is the actual ratio of the number of packets selected by a Selector from an input Population, to the Population Size. [ZsMD05] Please note that in this work the term **real sampling fraction** is used instead.

Hash-based selection: a filter specified by a hash domain, a hash function, and hash range and a hash selection range. [ZsMD05]

Time-based sampling: In time-based sampling the start and stop of the sampling interval is triggered by the time.[own definition but conforms to ZsMD05]

Count-based sampling: In count-based sampling the start and stop of the sampling interval is triggered by the packet count. [own definition but conforms to ZsMD05]

Systematic Sampling: Systematic sampling describes the process of selecting the starting points and the duration of the selection intervals according to a deterministic function. This can be for instance the periodic selection of every n^{th} element of a trace but also the selection of all packets that arrive at pre-defined points in time. Even if the selection process does not follow a periodic function (e.g., if the time between the sampling intervals varies over time) we consider this as systematic sampling as long as the selection is deterministic. [ZsMD05]

Parent population: The set of elements (e.g., packets) that contain all elements (e.g., all packets in the measurement interval. The size of the parent population denotes the number of elements and is usually by described with the letter N. [own definition, ZsMD05 uses a similar definition but more specific to packet selection]

Sample: A subset of elements, selected from the parent population. The sample size of the denotes the number of elements in the sample and is usually by described with the letter n. [own definition]

Population Size: The Population Size is the number of all packets in the Population. [ZsMD05]

Sample Size: The number of packets selected from the Population by a Selector. [ZsMD05]

Packet sampling: In packet sampling the basis element is a packet. All observed packets (e.g., in one measurement interval) form the parent population That mean in packet sampling some packets are selected out of all packets in the parent population. [own definition]

Flow sampling: In flow sampling the basis element is a flow. In flow sampling some flows are selected out of all flows in the parent population (e.g., all flows that are captured). Also mixtures of packet and flow sampling exit, where a clear differentiation is not possible. [own definition]

Random Sampling: Random sampling selects the starting points of the sampling intervals in accordance to a random process. The selection of elements are independent experiments. With this, unbiased estimations can be achieved. In contrast to systematic sampling, random sampling requires the generation of random numbers. [ZsMD05]

n-out-of-N sampling: In n-out-of-N sampling n elements are selected out of the parent population that consists of N elements. One example would be to generate random numbers and select all packets which have a packet position equal to one of the random numbers. For this kind of sampling the sample size is fixed. [ZsMD05]

Probabilistic sampling: In probabilistic sampling the decision whether an element is selected or not is made in accordance to a pre-defined selection probability. An example would be to flip a coin for each packet and select all packets for which the coin showed the head. For this kind of sampling the sample size can vary for different trials. The selection probability is not necessarily the same for each packet. Therefore we distinguish between uniform probabilistic sampling and non-uniform probabilistic sampling. [ZsMD05]

Stratified sampling: Stratified sampling is a multistage sampling method, where the elements of the parent population are first classified in accordance to a stratification variable into so-called strata. Then random sampling is performed per stratum. The term is often used to describe sampling, where a block in time is used and packets are randomly selected from this. It is o.k. to call this stratification, because it groups the elements. Nevertheless a stratification gain can only be achieved if the stratification variable correlates with the survey variable. For volume measurement this would mean, a benefit from stratified sampling could be expected if the packet size correlates with the arrival time.

B Measurement Groups and Standardization Efforts

This section contains a description of the most significant research groups that deal with network measurements. Since measurement provides the basis for research there are many more groups that work on the development of tools and the performance of measurements.

The ***National Laboratory for Applied Network Research*** [NLANR] is funded by the National Science Foundation and resides at the University of California at San Diego (UCSD). The group originally provided technical support and coordination for the very high performance Backbone Network Service (vBNS) of the NSF but now has expanded its scope to services and support for other high-performance network service providers, such as [Internet-2], Next Generation Internet [NGI], and [STARTAP]. The measurement and analysis team of NLANR operates a large network of active and passive measurement points.

The ***Cooperative Association for Internet Data Analysis*** [CAIDA] provides measurement and data analysis tools to support network operators. CAIDA originated from NLANR in 1997 and resides at the same location. While NLANR supports research and educational networks, CAIDA focuses on the commercial sector. The goal of the group is to enhance the cooperation among different groups and support them in maintaining a robust, scalable global Internet infrastructure. The group is well recognized in the measurement research community and members of the group publish their findings regularly at major conferences.

At the ***International Computer Science Institute*** [ICSI] the ICSI Center for Internet Research [ICIR] works on Internet Distance Maps [IDMAPS], measurement-based intrusion detection ([BRO], [Paxs99]) and secure measurement configuration for the measurement infrastructure NIMI [PaAM00]. The group is also quite active in standardization.

The ***Waikato Network Research Group*** [WAND] at University of Waikato Computer Science Department, is well known for their research on high-speed measurement hardware and developed the widely used DAG measurement boards [DAG]. The ***RIPE Network Coordination Center*** (RIPE NCC) [RIPE] is known for their active measurement box which is deployed in many research networks.

Further research groups come from large network operators (e.g., Sprint [IPMON], AT&T [ATTRD]) and vendors of network equipment (e.g., Cisco, Juniper, NEC, Hitachi). Also most national research and education networks have own measurement groups which more or less contribute to measurement research and standardization. Some research is also done by commercial measurement tool developer like HP, Agilent, and InMon.

Various groups within the [IETF], [IRTF] and [ITU-T] deal with standardization of different measurement aspects (e.g., [IPPM], [IMRG], [IPFIX], [PSAMP], [RMON]).

C Acronyms

AAA	Authentication, Authorization and Accounting
ACF	Autocorrelation Function
BGP	Border Gateway Protocol
BPF	Berkley Packet Filter
CI	Confidence Interval
CLR	Cell Loss Ratio
CTD	Cell Transfer Delay
DiffServ	Differentiated Services
DoS	Denial of Service
FDDI	Fiber Distributed Data Interface
GPS	Global Positioning System
GSL	GNU Scientific Library
i.i.d.	independent and identically distributed
ICMP	Internet Control Message Protocol
ID	Identification
IETF	Internet Engineering Task Force
IntServ	Integrated Services
IP	Internet Protocol
IPDV	IP packet delay variation
IPFIX	IP Flow Information Export
IQR	Inter Quartile Range
IRTF	Internet Research Task Force
ISP	Internet Service Provider
ITU-T	International Telecommunications Union, Telecommunication Standardization Sector
LAN	Local Area Network
LRD	Long Range Dependencies
MC	Multicast
MI	Measurement Interval
MIB	Management Information Bases
MTU	Maximum Transmission Unit
NAT	Network Address Translation
NSFNET	National Science Foundation Network

NTP	Network Time Protocol
OWD	One-way Delay
PSAMP	Packet Sampling
QoS	Quality of Service
RFC	Request for Comments (IETF/IRTF Documents)
RTFM	Real-time Traffic Flow Measurement
RTP	Real-time Transport Protocol
RTT	Round-trip Time
RV	Random Variable
SCBF	Space-Code Bloom Filter
SCTP	Stream Control Transmission Protocol
SLA	Service Level Agreement
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
TOS	Type of Service (field in IP header)
TTL	Time To Live (field in IP header)
UDP	User Datagram Protocol
WLAN	Wireless Local Area Network

D Mathematical Notation

The following table shows the mathematical notation used within this document.

Notation	Meaning
General Notation	
Θ	General notation for an (arbitrary) parameter of the parent population
$\hat{\Theta}$	Estimate for the parameter Θ. A "hat" above a parameter always denotes an estimate of the parameter.
$E\left[\hat{\Theta}\right]$	Expectation of $\hat{\Theta}$
$V\left[\hat{\Theta}\right]$	Variance of $\hat{\Theta}$
$StdErr\left[\hat{\Theta}\right] = StdErr_{abs}\left[\hat{\Theta}\right]$	Absolute standard error of $\hat{\Theta}$ (if no index is given the absolute error is meant)
$VarCoeff\left[\hat{\Theta}\right] = StdErr_{rel}\left[\hat{\Theta}\right]$	Relative standard error of $\hat{\Theta}$, coefficient of variation,
$Prob(A)$	Probability of A
z_c	Critical value (for normal distribution)
$1-\alpha$	Confidence level
ε	Absolute estimation error
N	Number of elements in the parent population. Here usually the number of all packets in the measurement interval.
n	Number of elements in the sample (sample size). Here usually the number of all selected packets from the measurement interval.
N_f	Number of packets from flow f in parent population
n_f	Number of packets from flow f in the sample
f_R	Real (attained) selection fraction
$f_{R,f}$	Real (attained) selection fraction for flow f
f_T	Target (configured) selection fraction
n_T	Target sample size
n_R	Real sample size
X, Y, Z	Random variables
x_i	Property of i^{th} packet (e.g., size in bytes, SLA conformance)

$x_{i,f}$	Property of i^{th} packet from flow f
R	Number of sampling runs
Volume Estimation	
Sum	Overall volume, sum of bytes from all observed packets in the measurement interval
$\hat{Sum}$	Estimated overall volume
Sum_f	Flow volume, sum of bytes from all observed packets from flow f in the measurement interval
$\hat{Sum}_f$	Estimated flow volume
μ_x	(empirical) mean of the parent population, here usually the mean packet size
μ_{x_f}	mean packet size for flow f
$\overline{x}$	mean packet size of all packets in sample
$\overline{x}_f$	mean packet size of all packets from flow f in sample
$s^2_{x_f}$	Variance of packet sizes of all packets in sample
$s^2_{x_f}$	Variance of packet sizes of all packets from flow f in sample
σ_x	(empirical) standard deviation of the parent population, here usually the standard deviation of packet sizes
σ_{x_f}	standard deviation of packet sizes for flow f
σ^2_x	(empirical) variance of the parent population, here usually the variance of the packet sizes
$\sigma^2_{x_f}$	Variance of packet sizes for flow f
$P_f = \frac{N_f}{N}$	Proportion of packets from flow f in measurement interval
b_{min}	Minimum packet size in bytes
b_{max}	Maximum packet size in bytes
Stratified Sampling	
K	Sampling period, number of packets in a subinterval (used for systematic and stratified sampling)
k	Number of selected packets from subinterval
L	Number of strata, e.g., number of subintervals in a measurement interval
l	l^{th} stratum
N_l	Number of elements in stratum l

n_l	Number of selected elements from stratum l
σ_l^2	Variance of the survey variable (packet sizes) in stratum l
$K_{f,l}$	Number of packets from flow f in stratum l
$k_{f,l}$	Number of selected packets from flow f in stratum l
μ_i	Mean packet size for i^{th} measurement interval
μ_i'	Predicted mean packet size for i^{th} measurement interval
Proportion Estimation	
M	Number of hits, here number of non-conformant packets in measurement interval
m	Number of non-conformant packets in sample
P	Proportion of number of non-conformant packets in measurement interval
d	Packet delay
d_{max}	Maximum packet delay
Δ_V , Δ_{SE}	Stratification gain
P_i'	Predicted proportion of violators for the i^{th} measurement interval

Table D-1: Mathematical Notation

E Derivation of Expectation and Variance for Volume Estimation

For the derivation of the expectation and variance of the flow volume a few standard formulas are needed, which are introduced here first (see e.g., [Rinn97]):

1) The definition of the expectation of a discrete random variable is given by

$$E\left[g(x)\right] := \sum_i g(x_i) \cdot Prob_i \quad \text{with} \quad Prob_i = Prob\left(X = x_i\right) \tag{E.1}$$

therefore

$$E\left[X^2\right] := \sum_i x_i^2 \cdot Prob_i \tag{E.2}$$

2) The variance of a random variable X can be expressed as

$$V[X] = E\left[X^2\right] - E[X]^2 \tag{E.3}$$

3) The expectation of a sum of independent random variables is given by the sum of the expectation values of the addends

$$E\left[\sum_i^k X_i\right] = \sum_i^k E[X_i] \tag{E.4}$$

if all X_i have the same expectation, one gets

$$E\left[\sum_i^k X_i\right] = k \cdot E[X_i] \tag{E.5}$$

4) The variance of a sum of independent random variables is given by the sum of the variances of the addends

$$V\left[\sum_i^k X_i\right] = \sum_i^k V[X_i] \tag{E.6}$$

if all X_i have the same variance, one gets

$$V\left[\sum_i^k X_i\right] = k \cdot V[X_i] \tag{E.7}$$

5) The law of the total probability:

$$Prob\left(Z\right) = \sum_{k=0}^{K} Prob\left(Z \mid Y = k\right) \cdot Prob\left(Y = k\right) \tag{E.8}$$

In order to derive the variance for the random variable Z first it is assumed that the variable Y is a fixed constant $y=k$. The random variable Z_k then is simply a sum of k random variables $X_1, ..., X_k$.

$$Z_k = \sum_{i=1}^{k} X_i \tag{E.9}$$

The probability function $Prob(Z_k)$ equals the conditional probability $Prob(Z|Y=k)$

$$Prob\left(Z_k\right) = Prob\left(Z \mid Y = k\right) \tag{E.10}$$

The probability function of Z can be calculated from $Prob(Z_k)$ with the law of the total probability:

$$Prob(Z) = \sum_k Prob(Z \mid Y = k) \cdot Prob(Y = k) \tag{E.11}$$

where K is the maximum number that the random variable Y can take. The variance of the discrete random variable Z is defined as:

$$V[Z] = E[Z^2] - E[Z]^2 \tag{E.12}$$

The expectation of Z can be calculated as follows:

$$E[Z] = \sum_j z_j \cdot Prob(Z = z_j) = \sum_j \left(z_j \cdot \sum_k Prob(Z = z_j \mid Y = k) \cdot Prob(Y = k) \right) \tag{E.13}$$

Since the sums are finite, they can be exchanged:

$$E[Z] = \sum_k \sum_j z_j \cdot Prob(Z = z_j \mid Y = k) \cdot Prob(Y = k) \tag{E.14}$$

With

$$E[Z \mid Y = k] = \sum_j z_j \cdot Prob(Z = z_j \mid Y = k) \tag{E.15}$$

and

$$E[Z \mid Y = k] = k \cdot E[X] \tag{E.16}$$

one gets

$$E[Z] = \sum_k Prob(Y = k) \cdot k \cdot E[X] \tag{E.17}$$

and with

$$E[Y] = \sum_k k \cdot Prob(Y = k) \tag{E.18}$$

the expectation of Z can be derived (see also [Fisz63]):

$$E[Z] = E[X] \cdot E[Y] \tag{E.19}$$

The expectation of Z^2 can be calculated as follows:

$$E[Z^2] := \sum_j z_j^2 \cdot Prob(Z = z_j) = \sum_j \left(z_j^2 \cdot \sum_k Prob(Z = z_j \mid Y = k) \cdot Prob(Y = k) \right) \tag{E.20}$$

Again the sums can be exchanged

$$E[Z^2] = \sum_k \sum_j z_j^2 \cdot Prob(Z = z_j \mid Y = k) \cdot Prob(Y = k) \tag{E.21}$$

With

$$E[Z^2 \mid Y = k] = \sum_j z_j^2 \cdot Prob(Z = z_j \mid Y = k) \tag{E.22}$$

one gets

$$E[Z^2] := \sum_k Prob(Y = k) \cdot E[Z^2 \mid Y = k] \tag{E.23}$$

With the definition of the variance of Z under the condition that $Y=k$

$$V\left[Z \mid Y=k\right]=E\left[Z^2 \mid Y=k\right]-E\left[Z \mid Y=k\right]^2 \tag{E.24}$$

one can substitute the expectation by

$$E\left[Z^2 \mid Y=k\right]=V\left[Z \mid Y=k\right]+E\left[Z \mid Y=k\right]^2 \tag{E.25}$$

and gets

$$E\left[Z^2\right]:=\sum_k Prob\left(Y=k\right)\cdot\left(V\left[Z \mid Y=k\right]+E\left[Z \mid Y=k\right]^2\right) \tag{E.26}$$

With

$$E\left[Z \mid Y=k\right]=k\cdot E\left[X\right] \tag{E.27}$$

and

$$V\left[Z \mid Y=k\right]=k\cdot V\left[X\right] \tag{E.28}$$

one gets

$$E\left[Z^2\right]:=\sum_k Prob\left(Y=k\right)\cdot k\cdot V\left[X\right]+Prob\left(Y=k\right)\cdot k^2\cdot E\left[X\right]^2 \tag{E.29}$$

With

$$E\left[Y\right]:=\sum_k k\cdot Prob\left(Y=k\right) \tag{E.30}$$

and

$$E\left[Y^2\right]:=\sum_{k=0}^{K} k^2\cdot Prob\left(Y=k\right) \tag{E.31}$$

the expectation of Z^2 can be derived:

$$E\left[Z^2\right]=E\left[Y\right]\cdot V\left[X\right]+E\left[Y^2\right]\cdot E\left[X\right]^2 \tag{E.32}$$

If now $E[Z]$ and $E[Z^2]$ is inserted, the variance of Z can be calculated as follows:

$$\begin{aligned} V\left[Z\right] &= E\left[Z^2\right]-E\left[Z\right]^2 \\ &= E\left[Y\right]\cdot V\left[X\right]+E\left[Y^2\right]\cdot E\left[X\right]^2-E\left[X\right]^2\cdot E\left[Y\right]^2 \\ &= E\left[Y\right]\cdot V\left[X\right]+E\left[X\right]^2\cdot\left(E\left[Y^2\right]-E\left[Y\right]^2\right) \end{aligned} \tag{E.33}$$

This results in the following formula for the variance of Z:

$$V\left[Z\right]=E\left[Y\right]\cdot V\left[X\right]+E\left[X\right]^2\cdot V\left[Y\right] \tag{E.34}$$

A derivation for continuous random variables can be found in [WeOw75].

F Derivation of Expectations and Variances for Proportion Estimation

Derivation of expectation and variance for estimate $\hat{P}$ for n-out-of-N sampling:

Expectation and variance of hyper geometric random variable *m*:

$$E[m] = n_T \cdot P \tag{F.1}$$

$$V[m] = \frac{N - n_T}{N - 1} \cdot n_T \cdot P \cdot (1 - P) \tag{F.2}$$

Expectation and variance of estimate $\hat{P}$:

$$E\left[\hat{P}\right] = E\left[\frac{m}{n_T}\right] = \frac{1}{n_T} \cdot E[m] = \frac{1}{n_T} \cdot n_T \cdot P = P \;\rightarrow \text{unbiased} \tag{F.3}$$

$$\begin{aligned} V\left[\hat{P}\right] &= V\left[\frac{m}{n_T}\right] = \frac{1}{n_T^2} \cdot V[m] = \frac{1}{n_T^2} \cdot \frac{N - n_T}{N - 1} \cdot n_T \cdot P \cdot (1 - P) \\ &= \frac{1}{n_T} \cdot \frac{N - n_T}{N - 1} \cdot P \cdot (1 - P) \end{aligned} \tag{F.4}$$

With $\frac{N - n_T}{N - 1} \approx \frac{N - n_T}{N} = 1 - \frac{n_T}{N} = 1 - f_T$ the relative standard error can be derived as follows:

$$StdErr_{rel}[\hat{P}] = \frac{1}{P} \cdot \sqrt{\frac{P \cdot (1 - P)}{n_T}} \cdot \sqrt{1 - \frac{n_T}{N}} = \sqrt{\frac{(1 - P)}{P \cdot n_T} \cdot \left(1 - \frac{n_T}{N}\right)} = \sqrt{\frac{(1 - P) \cdot (1 - f_T)}{P \cdot n_T}} \tag{F.5}$$

Variance for simplified (approximated by binomial distribution) n-out-of-N:

$$V\left[\hat{P}\right] = V\left[\frac{m}{n_T}\right] = \frac{1}{n_T^2} \cdot V[m] = \frac{1}{n_T^2} \cdot n_T \cdot P \cdot (1 - P) = \frac{P \cdot (1 - P)}{n_T} \tag{F.6}$$

$$StdErr_{rel}[\hat{P}] = \frac{1}{P} \cdot \sqrt{\frac{P \cdot (1 - P)}{n_T}} = \sqrt{\frac{(1 - P)}{P \cdot n_T}} \tag{F.7}$$

Derivation of expectation and variance for probabilistic sampling in accordance to [DuLT02]:

Estimate for the number of violators:

$$\hat{M} = \frac{1}{f_T} \cdot m = \frac{1}{f_T} \cdot \sum_{i=1}^{M} \omega_i \tag{F.8}$$

Expectation and variance calculated in accordance to [DuLT02] (neglecting the variability of n_R):

$$E\left[\hat{M}\right] = E\left[\frac{1}{f_T} \cdot \sum_{i=1}^{M} \omega_i\right] = \frac{1}{f_T} \cdot \sum_{i=1}^{M} E[\omega_i] = \frac{1}{f_T} \cdot M \cdot E[\omega_i] = \frac{1}{f_T} \cdot M \cdot f_T = M \;\rightarrow \text{unbiased} \tag{F.9}$$

$$V\left[\hat{M}\right] = V\left[\frac{1}{f_T}\cdot\sum_{i=1}^{M}\omega_i\right] = \frac{1}{f_T^2}\cdot\sum_{i=1}^{M}V\left[\omega_i\right] = \frac{1}{f_T^2}\cdot M\cdot V\left[\omega_i\right]$$
$$= \frac{1}{f_T^2}\cdot M\cdot f_T\cdot\left(1-f_T\right) = \frac{1}{f_T}\cdot M\cdot\left(1-f_T\right) = M\cdot\left(\frac{1}{f_T}-1\right) \quad \text{(F.10)}$$

Derivation of expectation and variance for estimate $\hat{P}$:

$$E\left[\hat{P}\right] = E\left[\frac{\hat{M}}{N}\right] = \frac{1}{N}\cdot E\left[\hat{M}\right] = \frac{M}{N} = P \rightarrow \text{unbiased} \quad \text{(F.11)}$$

$$V\left[\hat{P}\right] = V\left[\frac{\hat{M}}{N}\right] = \frac{1}{N^2}\cdot V\left[\hat{M}\right] = \frac{M}{N^2}\cdot\left(\frac{1}{f_T}-1\right) = \frac{P}{N}\cdot\left(\frac{1}{f_T}-1\right) \quad \text{(F.12)}$$

Derivation of relative standard error:

$$StdErr_{rel}\left[\hat{P}\right] = \frac{\sqrt{V\left[\hat{P}\right]}}{P} = \frac{\sqrt{\frac{P}{N}\cdot\left(\frac{1}{f_T}-1\right)}}{P} = \sqrt{\frac{1}{N\cdot P}\cdot\left(\frac{1}{f_T}-1\right)} = \sqrt{\frac{1}{P}\cdot\left(\frac{1}{n_T}-\frac{1}{N}\right)}$$
$$= \sqrt{\frac{N-n_T}{P\cdot n_T\cdot N}} = \sqrt{\frac{1-\frac{n_T}{N}}{P\cdot n_T}} = \sqrt{\frac{1-f_T}{P\cdot n_T}} \quad \text{(F.13)}$$

G Table of NZIX1 Experiments

The table shows experiments performed with trace: NZIX 20000706_120000. Classification S24D24 stands for a classification with source and destination network with netmask 0xFFFFFF00. S24D00 stands for source network only (with the same netmask).

Method	Classification	Sample Fraction	MI Length	Runs	Exp_ID	Result Table in Database
noAlg	S24D00	100	1,000,000	0	603	NZIX1_S24D00_noAlg_N1000k_f100_R0
n-out-of-N	S24D00	1	1,000,000	1000	643	NZIX1_S24D00_nofN_N1000k_f1_R1000
n-out-of-N	S24D00	2	1,000,000	1000	644	NZIX1_S24D00_nofN_N1000k_f2_R1000
n-out-of-N	S24D00	4	1,000,000	1000	645	NZIX1_S24D00_nofN_N1000k_f4_R1000
n-out-of-N	S24D00	5	1,000,000	1000	616	NZIX1_S24D00_nofN_N1000k_f5_R1000
n-out-of-N	S24D00	10	1,000,000	1000	646	NZIX1_S24D00_nofN_N1000k_f10_R1000
n-out-of-N	S24D00	20	1,000,000	1000	641	NZIX1_S24D00_nofN_N1000k_f20_R1000
n-out-of-N	S24D00	25	1,000,000	1000	648	NZIX1_S24D00_nofN_N1000k_f25_R1000
n-out-of-N	S24D00	50	1,000,000	1000	649	NZIX1_S24D00_nofN_N1000k_f50_R1000
1inK	S24D00	1	1,000,000	1000	636	NZIX1_S24D00_1inK_N1000k_f1_R1000
1inK	S24D00	2	1,000,000	1000	650	NZIX1_S24D00_1inK_N1000k_f2_R1000
1inK	S24D00	4	1,000,000	1000	638	NZIX1_S24D00_1inK_N1000k_f4_R1000
1inK	S24D00	5	1,000,000	1000	604	NZIX1_S24D00_1inK_N1000k_f5_R1000
1inK	S24D00	10	1,000,000	1000	640	NZIX1_S24D00_1inK_N1000k_f10_R1000
1inK	S24D00	20	1,000,000	1000	606	NZIX1_S24D00_1inK_N1000k_f20_R1000
1inK	S24D00	25	1,000,000	1000	622	NZIX1_S24D00_1inK_N1000k_f25_R1000
1inK	S24D00	50	1,000,000	1000	623	NZIX1_S24D00_1inK_N1000k_f50_R1000
systematic	S24D00	1	1,000,000	1000	625	NZIX1_S24D00_syst_N1000k_f1_R1000
systematic	S24D00	2	1,000,000	1000	626	NZIX1_S24D00_syst_N1000k_f2_R1000
systematic	S24D00	4	1,000,000	1000	628	NZIX1_S24D00_syst_N1000k_f4_R1000
systematic	S24D00	5	1,000,000	1000	605	NZIX1_S24D00_syst_N1000k_f5_R1000
systematic	S24D00	10	1,000,000	1000	629	NZIX1_S24D00_syst_N1000k_f10_R1000
systematic	S24D00	20	1,000,000	1000	607	NZIX1_S24D00_syst_N1000k_f20_R1000
systematic	S24D00	25	1,000,000	1000	631	NZIX1_S24D00_syst_N1000k_f25_R1000
systematic	S24D00	50	1,000,000	1000	633	NZIX1_S24D00_syst_N1000k_f50_R1000
noAlg	S24D24	100	1,000,000	0	593	NZIX1_S24D24_noAlg_N1000k_f100_R0
n-out-of-N	S24D24	5	1,000,000	1000	595	NZIX1_S24D24_nofN_N1000k_f5_R1000
1-in-K	S24D24	5	1,000,000	1000	594	NZIX1_S24D24_1inK_N1000k_f5_R1000
systematic	S24D24	5	1,000,000	1000	598	NZIX1_S24D24_syst_N1000k_f5_R1000

Table G-1: NZIX1 Experiments