Passive Internet Measurement: Overview and Guidelines based on Experiences

Wolfgang John, Sven Tafvelin and Tomas Olovsson

Department of Computer Science and Engineering
Chalmers University of Technology, Göteborg, Sweden

{firstname.lastname}@chalmers.se

Abstract

Due to its versatility, flexibility and fast development, the modern Internet is far from being well understood in its entirety. A good way to learn more about how the Internet functions is to collect and analyze real Internet traffic. This paper addresses several major challenges of Internet traffic monitoring, which is a prerequisite for performing traffic analysis. The issues discussed will eventually appear when planning to conduct passive measurements on high-speed network connections, such as Internet backbone links. After giving a brief summary of general network measurement approaches, a detailed overview of different design options and important considerations for backbone measurements is given. The challenges are discussed in order of their chronological appearance: First, a number of legal and ethical issues have to be sorted out with legislators and network operators, followed by operational difficulties that need to be solved. Once these legal and operational obstacles have been overcome, a third challenge is given by various technical difficulties when actually measuring high-speed links. Technical issues range from handling the vast amounts of network data to timing and synchronization issues. Policies regarding public availability of network data need to be established once data is successfully collected. Finally, a successful Internet measurement project is described by addressing the aforementioned issues, providing concrete lessons learned based on experiences. As a result, the paper presents tutorial guidelines for setting up and performing passive Internet measurements.

1 Introduction

The usage of the Internet has changed dramatically since its initial operation in the early-80s, when it was a research project connecting a handful of computers, facilitating a small set of remote operations. Nowadays (2009), the Internet serves as the data backbone for all kinds of protocols, making it possible to exchange not only text, but also voice, audio, video and various other forms of digital data between hundreds of millions of nodes, ranging from traditional desktop computers, servers or supercomputers to all kinds of wireless devices, embedded systems, sensors and even home equipment.

Traditionally, an illustration of the protocol layers of the Internet has the shape of an hourglass, with a single Internet Protocol (IP) on the central network layer and an increasingly wider spectrum of protocols above and below. Since the introduction of IP in 1981, which is basically still unchanged, technology and protocols have developed significantly. Underlying transmission media evolved from copper to fiber optics and WIFI, routers and switches became more and more intelligent and are able to handle Gbit/s instead of Kbit/s and additional middleware boxes have been introduced (e.g. NAT and firewalls). But also above the network layer new applications have constantly been added, ranging from basic services such as DNS and HTTP, to recent, complex P2P protocols allowing applications such as file-sharing, video streaming and telephony. With IPv6, even the foundation of the Internet is finally about to be substituted. This multiplicity of protocols and technologies leads to an ongoing increase in complexity of the Internet as a whole. Of course, individual network protocols and infrastructures are usually well understood when tested in isolated lab environments or network simulations. However, their behavior when observed while interacting with the vast diversity of applications and technologies in the hostile Internet environment is often unclear, especially on global scale.

This lack of understanding is further amplified by the fact that the topology of the Internet was not planned in advance. It is the result of an uncontrolled extension process, where heterogeneous networks of independent organizations have been connected one by one to the main Internet (*INTERconnected NETworks*). This means that each autonomous system (AS) has its own set of usage and pricing policies, QoS measures and resulting traffic mix. Thus usage of Internet protocols and applications is not only changing with time, but also with geographical locations. As an example, Nelson et al. [1] reported about an unusual application mix on a campus uplink in New Zealand due to a restrictive pricing policy, probably caused by higher prices for trans-pacific network capacities at this time.

Finally, higher connectivity bandwidths and growing numbers of Internet users also lead to increased misuse and anomalous behavior [2]. Not only the numbers of malicious incidents keep rising, but also the level of sophistication of attack methods and tools has increased. Today, automated attack tools employ more and more advanced attack patterns and react on the deployment of firewalls and intrusion detection systems by clever obfuscation of their malicious intentions. Malicious activities range from scanning to more advanced attack types such as worms and various denial of service attacks. Even well-known or anticipated attack types reappear in modified variants, as shown by the recent renaissance of cache poisoning attacks [3]. Unfortunately, the Internet, initially meant to be a friendly place, eventually became a very hostile environment that needs to be studied continuously in order to develop suitable counter strategies.

Overall, this means that even though the Internet may be considered to be the most important modern communication platform, its behavior is not well understood. It is therefore crucial that the Internet community understands the nature and detailed behavior of modern Internet traffic, in order to be able to improve network applications, protocols and devices and protect its users.

The best way to acquire a better and more detailed understanding of the modern Internet is to monitor and analyze real Internet traffic. Unfortunately, the above described rapid development has left little time or resources to integrate measurement and analysis possibilities into Internet infrastructure, applications and protocols. To compensate for this lack, the research community has started to launch dedicated Internet measurement projects, usually associated with considerable investment of both time and money. However, the experiences from a successful measurement project showed that measuring large-scale Internet traffic is not simple and involves a number of challenging tasks. In order to help future measurement projects to save some of their initial time expenses, this paper gives an overview of the major challenges which will eventually appear when planning to conduct measurements on high-speed network connections. Experiences from the MonNet project will then provide guidelines based on lessons learned (Section 8).

1.1 How to read this paper

Section 2 gives an overview of different network traffic measurement approaches and methodologies. Sections 3-7 address the main challenges encountered while conducting passive Internet measurements. The challenges are discussed in order of their chronological appearance: First, a number of legal and ethical issues have to be sorted out with legislators and network operators before data collection can be started (Sections 3 and 4). Second, operational difficulties need to be solved (Section 5) such as access privileges to the network operator's premises. Once legal and operational obstacles are overcome, a third challenge is given by various technical difficulties when actually measuring high-speed links (Section 6), ranging from handling of vast data amounts to timing issues. Next public availability of network data are discussed, which should eventually be considered once data are successfully collected (Section 7). Section 8 then outlines the MonNet project, which is the measurement project providing the experience for the present paper. Each point from Sections 3 - 7 will be revisited and the specific problems and solutions as experienced in the MonNet project are presented. These considerations are then summarized presenting the most important lessons learned in each particular section, providing a quick guide for future measurement projects. Finally, Section 9 discusses future challenges of Internet measurement and concludes the paper.

2 Overview of network measurement methodologies

This section gives an overview of general network measurement approaches. The basic approaches are categorized among different axes and the most suitable methods for passive Internet measurements according to current best practice are pointed out.

The most common way to classify traffic measurement methods is to distinguish between **active** and **passive** approaches. Active measurement involves injection of traffic into the network in order to probe certain network devices (e.g. PING) or to measure network properties such as round-trip-times (RTT) (e.g. traceroute), one-way delay and maximum bandwidth. Pure observation of network traffic, referred to as passive measurement or monitoring, is non-intrusive and does not change the existing traffic. Network traffic is tapped at a specific location and can then be recorded and processed at different levels of granularity, from complete packet-level traces to statistical figures. Even if active measurement offers some possibilities that passive approaches cannot provide, in this paper only passive measurement is considered, which is best suitable for analysis of Internet backbone traffic properties.

Passive traffic measurement methods can be further divided into **software-based** and **hardware-based** approaches. Software-based tools modify operating systems and device drivers on network hosts in order to obtain copies of network packets (e.g. BSD packet filter [4]). While this approach is inexpensive and offers good adaptability, its possibilities to measure traffic on high-speed networks are limited [5]. In contrast, hardware-based methods are designed specifically for collection and processing of network traffic on high-speed links such as an Internet backbone. Special traffic acquisition hardware collects traffic directly on the physical links (e.g. by using optical splitters) or on network interfaces (e.g. mirrored router ports). Since highly specialized, such equipment is rather expensive and offers limited versatility.

Once network data are collected, it needs to be processed to fulfill its particular purpose.Traffic processing can be done **online**, **offline** or in a combination of both approaches. Online processing refers to immediate processing of network data in "real time", which is essential for applications such as traffic filters or intrusion detection systems. Sometimes only parts of the data processing are done online, as typically done when collecting condensed traffic statistics or flow-level summaries. Offline processing on the other hand is performed on network data after it is stored on a data medium. Offline processing is not time critical and offers the possibility to correlate network traffic collected at different times or different locations. Furthermore, stored network data can be re-analyzed with different perspectives over and over again. These advantages make offline processing a good choice for complex and time consuming Internet analysis.

Internet measurement can furthermore operate on different **protocol layers**, following the Internet reference model [6]. While link-layer protocols dictate the technology used for the data collection (e.g. SONET/HDLC, Ethernet), one of the most studied protocols is naturally the Internet Protocol (IP), located on the network layer. The Internet measurement community also shows great interest in the analysis of transport layer protocols, especially TCP and UDP. Some Internet measurement projects even have the possibilities to study all layers, including application layer protocols. In practice, most measurement projects consider mainly network and transport layer protocols due to privacy and legal concerns, as discussed later (Sections 3 and 4)

Data gathered on different protocol layers can present different levels of granularity. The most coarse granularity is provided by cumulated **traffic summaries and statistics**, such as packet counts or data volumes, as typically provided by SNMP [7]. Another common practice is to condense network data into **network flows**. A flow can be described as a sequence of packets exchanged between common endpoints, defined by certain fields within network and transport headers. Instead of recording each individual packet, flow records are stored, containing relevant information about the specific flow. Such flow records can be unidirectional, as in the case of NetFlow [8], or bidirectional, as used in different studies by MonNet [9, 10, 11]. The finest grained level of granularity is provided by **packet-level traces**. Packet-level traces can include all information of each packet observed on a specific host or link. While such **complete packet-level traces** offer the best analysis possibilities, they come along with a number of technical and legal issues, as discussed in Chapters 3 - 6. It is therefore common practice to reduce the stored information to packet headers up to a certain protocol level, e.g. including network and transport

protocols only, as done for the MonNet traces.

Finally, packet-level network traces can be stored in different trace formats. Unfortunately, there is no standardized trace format, so developers of trace collection tools historically defined their own trace formats. The most popular trace format, especially common for traces from local area networks (LANs), is the **PCAP format**, the format of the BSD Packet Filter and TCPdump. For traces of wide area networks (WANs), an often used format was defined by Endace, the Endace record format (ERF), formerly also known as **DAG format**. Other trace formats seen in the Internet measurement community include CAIDA's CORALReef format CRL [12] or NLANR's formats FR, FR+ and TSH. This diverseness in trace formats introduces some problems, since publicly available analysis tools usually do not recognize all of these formats, making conversion of traces from one format to another necessary. Since PCAP can be seen as the de-facto standard, almost all conversion tools are able to convert their own format to or from this format. Conversion, however, is usually not without cost. Different timestamp conventions within the trace formats often lead to loss of timestamp precision, which should be considered when performing timing sensitive operations.

3 Legal background

In this section the legal background of Internet measurement is presented, which is somewhat in contrast to actual political developments and common academic practice. Current laws and regulations on electronic communication rarely explicitly consider or mention the recording or measurement of traffic for research purposes, which leaves scientific Internet measurement in some kind of legal limbo. In the following paragraphs the existing regulations for the EU and the US are briefly outlined in order to illustrate the legal complications network research is struggling with.

3.1 European Union (EU) directives

Privacy and protection of personal data in electronic communication in EU countries are regulated by the *Directive 95/46/EC on the protection of personal data* [13] of 1995 and the complementing *Directive 2002/58/EC on Privacy and Electronic Communications* [14] of 2002. Data retention regulations have recently been further amended with the *Directive 2006/24/EC on the retention of data generated or processed in electronic communication* [15].

The Data protection directive (Directive 95/46/EC) defines personal data in Article 2a as *"any information relating to an identified or identifiable natural person (data subject)"*. Besides names, addresses or credit card numbers, this definition thereby also includes email and IP addresses. Furthermore, data are defined as personal as soon as someone can potentially link the information to a person, where this someone not necessarily needs to be the one possessing the data. Processing of personal data are then defined in Article 2b as *"any operation or set of operations which is performed upon personal data, whether or not by automatic means, such as ... collection, recording, ...storage, ..."*, which means that Internet traffic measurement clearly falls into the scope of this directive. Summarized, Directive 95/46/EC defines conditions under which the processing of personal data are lawful. Data processing is e.g. legitimate with consent of the user, for a task of public interest or for compliance with legal obligations (Article 7). Further conditions include the users (or "data subjects") right for transparency of the data processing activities (Articles 10 and 11), the users right of access to own personal data (Article 12) and principles relating to data quality (Article 6). The latter describes that data are only allowed to be processed for specified, explicit and legitimate purposes. However, further processing or storage of personal data for historical, statistical or scientific purposes is not incompatible with these conditions, as long as appropriate safeguards for this data are provided by individual member states.

The e-privacy directive (Directive 2002/58/EC) complements the data protection directive of 1995, targeting matters which have not been covered earlier. The main subject of this directive is *"the protection of privacy in the electronic communication sector"*, which was required to be updated in order to react on requirements of the fast changing digital age. In contrast to the data protection directive, the e-privacy directive is not only applied to natural but also to legal persons. Besides dealing with issues like treatment of spam or cookies, this directive also includes regulations concerning confidentiality of information and treatment of traffic data. Some of the regulations are especially relevant for Internet

measurement. Specifically, Article 5 states that *"listening, tapping, storage or other kinds of interception or surveillance of communications and the related traffic data by persons other than users"* are prohibited, with the exception of given consent by the user or the necessity of measures in order *"to safeguard national security, defense, public security, and the prevention, investigation, detection and prosecution of criminal offenses"* (Article 15(1)). Furthermore, Article 6(1) obliges service providers to erase or anonymize traffic data when no longer needed for transmission or other technical purposes (e.g. billing, provision, etc.), again with the only exception of national security issues (Article 15(1)).

The data retention directive (Directive 2006/24/EC) was among others a reaction on recent terrorist attacks (i.e. July 2005 in London), requiring communication providers to retain connection data for a period of between 6 months and 2 years *"for the purpose of the investigation, detection and prosecution of serious crime""* (Article 1). When this directive was released in March 2006, only 3 EU countries had legal data retention in force. The remaining countries declared to postpone application of this directive regarding Internet access, Internet telephony and Internet email, which was possible until 14 March 2009 according to Article 15(3). At present (October 2009) 22 of the 27 EU countries have transposed the directive (at least partial, as in the case of Luxembourg and the UK) by implementing the different national laws. The remaining five countries (i.e. Austria, Greece, Ireland, Poland, and Sweden) have yet failed to install national laws following the directive. An updated overview of national data retention policies and laws can be found online at "Vorratsdatenspeicherung.de" [16].

For current measurement projects in EU countries these directives basically say that Internet traffic measurement for scientific purposes requires user consent, since such projects are not subject of national security. User content could e.g. be obtained by adding a suitable passage to the "Terms of Service" signed by network users. Additionally, any individual member state has the possibility to permit Internet measurement for scientific purposes if appropriate safeguards are provided. With the introduction of the data retention directive, providers are legally required to store connection data. However, in order to be able to actually execute this directive, a number of technical challenges need to be solved first (Section 6). Experiences and lessons learned from scientific Internet measurement projects are therefore vital and further underline the relevance of Internet measurement.

3.2 United States (US) laws

In contrast to the EU, privacy in the US is handled by a patchwork of case law, state and federal industry-specific laws [17]. The overview of US privacy laws in the present paper will follow a recent article by Sicker et al. [18], thereby focusing on federal laws of the US only (as opposed to state laws), especially since they are probably best compared to the overarching EU directives. There are two relevant sets of federal US laws applying to Internet measurement: one for real-time monitoring, and one for access to stored data.

When monitoring network traffic in real time, US laws distinguish between monitoring of user content and non-content such as header data. Real-time content monitoring is regulated by the *Wiretap Act* (18 U.S.C. §2511 [19]), basically stating that interception of communications is prohibited. There are, however, some exceptions to this basic rule, including user consent of at least one party of the communication as well as the providers' right to protect their networks and to help tracking culprits. Real-time monitoring of non-content (i.e. header data) was unregulated in the US until 2001, when the 9/11 attacks lead to the USA PATRIOT Act. This law amended the *Pen Register and Trap and Trace Act* (18 U.S.C. §3127 [20]) in order to apply it to recording or capturing of *"dialing, routing, addressing, or signaling information"* in context of electronic communications, which clearly includes non-content such as packet headers and IP address information. Consequently, also recording of packet header traces is prohibited in the US since 2001. Again, user consent and provider monitoring are exceptions stated in the act.

Access to stored network data, i.e. sharing of data traces, is in US federal laws regulated by the *Electronic Communications Privacy Act* (18 U.S.C. §2701-§2703 [21, 22, 23]). Basically, it is prohibited for network providers to give away stored records of network activity, regardless whether or not they include user content. Besides the exception of user consent there are two further exceptions to this basic rule. First, this rule does not apply to non-public providers, which means that data collected at private companies or organizations can be shared with other organizations or researchers. Second, non-content

records (e.g. header traces) can be shared with anyone, with exception of the government. This leaves some uncertainty about the definition of "government entities", since scientific projects and researchers might be funded or co-sponsored by governmental money.

3.3 Scientific practice

For researchers it is not always obvious which regulations are in force. The borders between private and public networks as well as the difference between signaling or header data and user content is sometimes blurred and fuzzy, which makes it difficult to relate to the correct piece of law. This is especially true for amateurs in juristic matters, such as typical network scientists. Common privacy protection measures have been surveyed on datasets used in 57 recent Internet measurement related articles in [18], showing that a majority of network traces were collected on public networks and stored as packet headers only. Discussions about trace anonymization or the difference between content and non-content was brought up in very few articles, probably due to page restrictions. However, it can be assumed that most researchers are aware of their responsibility towards the users and are anxious about privacy concerns, as described in Section 4.

As pointed out by Sicker et al. [18], often there is a *"disconnect between the law and current academic practice"*. Even though leading researchers try to close the gap between Internet researchers and lawyers by clarifying common misconceptions about the Internet [24], laws are not likely to be changed in favor of scientific Internet measurement anytime soon. According to Sicker at al. [18], a first important step towards de-criminalization of Internet measurement could be a community-wide consensus about privacy-protecting strategies formulated in a public document. Furthermore, the authors present some basic strategies for protecting user privacy, ranging from the often impossible task of getting user consent (e.g. signed "Terms of Service") to traditional de-sensitization techniques such as anonymization and data reduction (see Sections 4 and 6.2). The network researcher's motto should first of all be: *Do no Harm!* Even though researchers might sometimes unavoidably operate in legal grey zones, it is likely that no legal prosecution will be started as long as careful measures to avoid privacy violations following "common sense" have been taken and no harm has been done.

In a recent paper, Kenneally and Claffy go one step further and propose the Privacy-Sensitive Sharing framework (PS2) [17], a framework supporting proactive management of privacy risks. The proposed model is a combination of a policy framework that satisfies obligations of both data seekers and data providers, and a technology framework able to enforce these obligations. As a result, PS2 should reveal that actual data sharing is less risky (in form of privacy risks) than *not* sharing data (and inability to understand and anticipate the Internet and its security threads), especially when considering the importance of modern Internet as an underlying, critical infrastructure for economical, professional, personal, and political life [25].

4 Ethical and moral considerations

Besides potential conflicts with legal regulations and directives, Internet measurement activities raise also moral and ethical questions when it comes to privacy and security concerns of individual users or organizations using the networks. These considerations include discussions about what to store, how long to store and in which ways to modify stored data. The goal is to fulfill privacy and security requirements of individuals and organizations, while still keeping scientific relevant information intact. Since network data can potentially compromise user privacy or reveal confidential network structures or activities of organizations, operators usually give permission to perform Internet measurement with at least one of the following restrictions:

1. *keep* raw measurement data *secret*;
2. *de-sensitize* the data, which can be done in one or both of the following ways:
   (a) *remove sensitive data (such as packet payload data)* in packet-level traces;
   (b) *anonymize or de-identify* packet traces and flow data.

De-sensitization refers to the process of removing sensitive information to ensure privacy and confidentiality. An example where un-desensitized measurement data are required would be network forensics conducted by governmental authorities. In this case data are kept secret, i.e. it is accessed by a limited number of trusted persons only. Within research projects, however, it is common that de-sensitization is required. Anonymization in this context refers to the process of removing or disguising information which reveals the real identity of communication entities. Some information, such as IP addresses, can be used to pinpoint individual users. This privacy threat makes IP address anonymization a common requirement even for measurements which are only kept internally, inside a network operator's organization.

The above stated de-sensitization actions, payload removal and anonymization, might satisfy both data providers (operators) and data seekers (researchers and developers) analyzing the data. There are, however, a number of detailed questions, which are not necessarily answered by often imprecise and broadly stated policies. Some important considerations are discussed below.

4.1 What to keep?

Even if it is decided to store packet header traces only, it is not always explicitly stated where user payload really starts. A common way to interpret "packet headers" is to keep IP and TCP (UDP) headers only, stripping off data after transport headers. However, one could argue that application headers are technically not user payload, and therefore could be kept as well. This may lead to problems in some cases (e.g. SMTP and HTTP headers), since a lot of sensitive information can be found there. Other application headers, such as SSH and HTTPS, violate no obvious privacy issues, assuming that IP address anonymization is done for all layers of packet headers. Furthermore, application headers introduce practical problems since the number of network applications is virtually infinite and not all applications use well defined headers. A solution is to store the first N bytes of the payload following transport protocols. Saving the initial bytes of packet payloads is sufficient for classifying traffic using signature matching (shown e.g. by Karagiannis et al.[26]) and offers a number of additional research possibilities, such as surveying frequency and type of packet encryption methods. Even if packets with privacy-sensitive application data (e.g. SMTP) would be treated differently and stored without any payload beyond transport layer, there is still a large degree of uncertainty left about how much sensitive information is included in unknown or undefined application payloads or malformed packets not recognizable for the processing application. This remaining uncertainty might be tolerable if traces are only accessed by a limited number of trusted researches, but is unsuitable for traces intended to become publicly available.

Even if the boundary between packet header and packet payload is clearly defined for most protocols (e.g. payload starts beyond transport layer), the researcher needs to decide how to treat unusual frames, not defined within most available trace processing tools. One such example is routing protocols such as CLNS routing updates (Connectionless Network Protocol) and CDP messages (Cisco Discovery Protocol). Even if routing information is not revealing privacy-sensitive data about individual users, it reveals important information about network layout and topology, which in turn can be important input to de-anonymization attacks. Another example are all kinds of unknown or malformed headers, which might not be recognized by processing tools, but still contain sensitive information following malicious packet headers [27]. Policies for how to treat this kind of packets include:

1. packet truncation by default after a specified number of bytes;
2. packet dropping (which should be recorded in the meta-data of the specific trace);
3. keeping the un-truncated packet (which might bear security and privacy risks).

Finally, privacy of datasets can be improved by removing network data from hosts with unique, easy distinguishable behavior, as suggested by Coull et al. in [28]. Such hosts can include DNS servers, popular HTTP or SMTP servers or scanning hosts. Obviously, this approach leaves a biased view of network traffic, which might be unsuitable for certain research purposes. It is therefore crucial that removal or special treatment of packets from specially exposed hosts is well documented and commented in the descriptions or the meta-data of the respective network traces.

4.2 How to anonymize?

If anonymization of network traces is required, it still needs to be decided which header fields to anonymize and how. Generally, it should be noted that "*anonymization of packet traces is about managing risk*", as pointed out by Pang et al. [29]. Datasets from smaller, local networks might be more sensitive than data from highly aggregated backbone links when it comes to attacks trying to infer confidential information such as network topologies or identification of single hosts. Coull et al. [28] also showed that hardware addresses in link-layer headers can reveal confidential information, which is a problem for Ethernet-based measurements, but not for Internet measurement on backbone links. Furthermore, the age of the datasets being published plays an important role since the Internet has a very short-lived nature, and network architectures and IP addresses change frequently and are hard to trace back. Generally, anonymization is an important measure to face privacy concerns of users, even though it needs to be noted that all proposed anonymization methods have been shown to be breakable to a certain degree, given an attacker with sufficient know-how, creativity and persistency [28, 30, 31, 32]. This was stated nicely by Allman and Paxson in [33], when saying that publisher of network traces "*are releasing more information than they think*"!

Currently, the most common practice to anonymize packet headers is to anonymize IP address information only, which is often sufficient for internal use (i.e. only results, but no datasets will be published). As discussed above, in some situations when traces are planned to be published, a more complete method is required, offering the possibility to modify each header and payload field with individual methods, including email addresses, URLs and usernames/passwords. Such a framework is publicly available and described by Pang et al. in [29]. However, how different fields are modified has to be decided by the researcher or agreed upon in anonymization policies. The increasing importance of data anonymization for the Internet measurement community has recently resulted in the organization of a dedicated workshop on Network data anonymization [34], which sets out to advance theory and practice of anonymization as it applies to network data.

4.2.1 Anonymization methods

In the following paragraphs, we list and discuss some common methods of how to anonymize the most sensitive information in packet headers, namely IP addresses. IP address anonymization is here defined as the irreversible mapping between the real and the anonymized IP addresses.

1. *One constant:* The most simple method is to substitute all IP addresses with one constant, which collapses the entire IP address space to one single constant with no information content. A refined version of this method is to keep the first N bits of addresses unmodified, and replace the remaining bits with a constant (e.g. set them to zero).

2. *Random permutation:* Another rather simple method is random permutation, which creates a one-to-one mapping between real and anonymized addresses. This method is only irreversible given a proper secrecy concerning the permutation table. Furthermore the subnet information implicitly included in the real addresses is lost.

3. *Pseudonymization:* The idea of random permutation is very similar to a method called pseudonymization, where each IP address is mapped to a pseudonym, which might or might not have the form of a valid IP address. It is only important that a one-to-one mapping is provided.

4. *Prefix-preserving anonymization:* A special variation of pseudonymization has the property of preserving prefix information, and is therefore referred to as prefix-preserving anonymization. A prefix-preserving anonymization scheme needs to be impossible or at least very difficult to reverse while maintaining network and subnet information, which is crucial for a many different types of analysis.

   (a) *TCPdpriv:* The first popular prefix-preserving anonymization technique was used in TCPdpriv, developed by Minshall in 1996 [35]. The prefix preserving anonymization function of TCPdpriv applies a table-driven translation based on pairs of real and anonymized IP addresses. When new translations are required, existing pairs are searched for the longest prefix match. The

first k bits matching the already translated prefix are then reused, and the remaining $32 - k$ bits are replaced with a pseudo-random number and the address is added to the table. The drawback of this approach is that the translations are inconsistent when used on different traces, since translation depends on the order of appearance of the IP addresses. This problem can be solved if translation tables are stored and reused. The approach, however, still leaves the problem that traces cannot be anonymized in parallel, which is desired practice when dealing with large volumes of Internet data.

(b) *Crypto-PAn:* The drawback of TCPdpriv was fixed by a Cryptography-based Prefix-preserving Anonymization method, Crypto-PAn, described by Xu et al. in 2002 [30]. Crypto-PAn offers the same prefix-preserving features as TCPdpriv, with the additional advantage of allowing distributed and parallel anonymization of traces. Instead of a table-driven approach, Crypto-PAn establishes a deterministic one-to-one mapping by use of a key and a symmetric block cipher. This anonymization key is the only information which needs to be copied when consistent anonymization is done in parallel. Crypto-Pan is nowadays probably the most widely used anonymization method, and has since been modified in order to suit specific requirements, such as anonymization of flow data [36] or online anonymization of traffic on 1 Gbit/s links [37].

4.2.2 Quality of anonymization

Recently, different successful attacks on IP addresses in anonymized traces have been presented [28, 31, 32, 38]. With the awareness of the weaknesses of anonymization methods, it is important to establish policies and agreements between data-receivers and data-sharer not to carry out de-anonymization attempts [17]. Furthermore, Pang et al. [29] argue that anonymizing IP addresses alone might not be enough to preserve privacy. Consequently, a framework which allows anonymization of each header field according to an anonymization policy was presented. They also propose a novel approach to IP address anonymization. External addresses are anonymized using the widely used Crypto-PAn, while internal addresses are mapped to unused prefixes in the external mapping. Note, however, that this scheme does not preserve prefix relationships between internal and external addresses, but is on the other hand less vulnerable to certain types of attacks, as noted by Coull et al. [28].

At present, however, Crypto-PAn is still widely used and sets an de-facto standard for trace anonymization. Thus proper handling of the anonymization key is another issue that needs to be taken care of by researchers. The key is crucial, because with knowledge of the key is it straight-forward to re-translate anonymized addresses bit by bit, which opens for a complete de-anonymization of the trace. The safest solution is to generate a new key for each trace anonymization procedure, which is destroyed immediately after the anonymization process. Obviously, this approach would not provide consistency between different anonymized traces, which is one of the main features of Crypto-PAn. It is possible to re-use a single key across traces taken on different times or locations. In such setups, access to this key needs to be highly restricted, and clear policies for scenarios involving duplication of the key (e.g. for parallel anonymization purposes) are required.

4.3 Temporary storage

After discussing different considerations regarding payload removal and anonymization, it is still an open question when these operations should be performed. If a policy or an agreement with the network operator states that network data are only allowed to be stored if it is payload-stripped and anonymized, does this mean that unprocessed traces are not allowed to be recorded on mass storage devices at all? If so, is there sufficient computational power to process potentially huge amounts of Internet traffic in "real time" during the collection process? And if temporary storage of raw-traces is necessary for processing purposes, how long does "temporary" really mean? Does the processing (payload removal and anonymization) need to be started immediately after finishing the collection? And how to proceed in case of processing errors, which might require manual inspection and treatment? When is it safe to finally delete unprocessed raw-traces? Such detailed questions are not always answered by existing policies, so it is often up to the researchers to make adequate, rational choices in order to minimize the risks of violating privacy and confidentiality concerns of users and organizations.

4.4 Access and security

Since network data can contain a number of sensitive and confidential information, it is crucial to prevent unauthorized access to (raw) trace data. In case where traces are regarded as very sensitive, it might even be necessary to encrypt the archived network data. If data needs to be copied, there should be clear hand-over policies, which help to keep track of the distribution of datasets. Additionally, the monitoring equipment and measurement nodes need to be secured carefully, since access to functional measurement nodes is probably an even better source to attackers than already collected traces. For measurement equipment and data the same security measures as for all sensitive data centers should be applied. Besides restricting physical access to facilities housing measurement equipment and storage, also network access needs to be strictly regulated and monitored. Finally, especially in case of discontinuous measurement campaigns, measurement times should be kept secret to minimize the risk of de-anonymization attacks involving identifiable activities during the measurement interval.

5 Operational difficulties

Data centers and similar facilities housing networking equipment are usually well secured and access rights are not granted easily, which is especially true for external, non-operational staff, such as researchers. Often it is required that authorized personnel are present when access to certain premises is needed. This dependency makes planning and coordination difficult and reduces flexibility and time-efficiency. Flexibility constraints are further exaggerated by the geographic location of some premises, since they are not necessarily situated in close proximity to the researchers institute. Moreover, some significant maintenance tasks, such as installation of optical splitters, require interruption of services, which is undesired by network operators.

The above indicated operational difficulties suggest the need of careful planning of measurement activities, including suitable risk management such as slack time and hardware redundancy when possible. Generally, the sparse on-site time should be utilized with care in order to disturb normal operations as little as possible. A good way of doing so is to use hardware with remote management features, providing maximum control of operating system and hardware of the installed measurement equipment. Such remote management capabilities should include possibilities to reset machines and offer access to the system console, independent from operating systems.

A final challenge in planning Internet measurements is the short-lived nature of network infrastructure, which might influence ongoing measurement projects depending on their specific measurement locations. Generally, measurements are carried out in a fast changing environment, including frequent modifications in network infrastructure and equipment but also changes in network topologies and layouts. This changeful nature of network infrastructure is especially cumbersome for projects intended to conduct longitudinal measurements. Some changes in network infrastructure might not only require modifications or replacement of measurement equipment, but also hamper unbiased comparison of historical data with contemporary measurement data.

6 Technical aspects

Measurement and analysis of Internet traffic is not only challenging in terms of legal and operational issues, but also it is above all a technical challenge. In the following subsections we first discuss methods to physically access and collect network traffic. We will then provide discussions about other important aspects regarding Internet measurement, including strategies to cope with the tremendous amounts of data and some considerations for how to get confidence in the measured data. Finally, we will discuss the important challenge of timing and synchronization. Timing is an important issue in network measurement, especially when timing-sensitive correlation of different traffic traces is required, such as passive one-way delay (OWD) measurements or when merging network traces measured on links of opposite direction.

6.1 Traffic access methods

On shared-medium protocols such as Ethernet, passive measurements can be carried out by all nodes connected to the medium via commodity network interface cards (NICs) running in promiscuous mode. Unfortunately, NICs are not designed for monitoring purposes and do not offer effective and precise

recording of network traffic (e.g. imprecise timestamp generation as discussed in 6.4.2 or unreported packet loss). In Ubik and Zejdl [5] it was shown that it is theoretically possible to monitor 10 Gbit/s links with commodity NICs (which currently can support up to 10 Gbit/s for Gigabit-Ethernet). This, however, comes with the cost of high CPU load[1] and the mentioned precision deficiencies.

Specialized traffic monitoring hardware on the other hand can provide precise traffic collection without putting extra CPU load on the monitoring host, which can then be used to perform online traffic processing instead. Currently, the most common capture cards for high-speed network measurements are Endace DAG cards [39], but also other companies offer such equipment, such as Napatech [40] or Invea-Tech [41]. Modern capture cards provide lossless, full packets data collection with precise timestamping and filtering capabilities for link speeds of up to 10 Gbit/s. These cards also report about collection problems such as dropped packets and checksum errors. Endace recently even released a capture box for 40 Gbit/s linespeed [42], which is essentially splitting 40 Gbit/s input into 4 x 10 Gbit/s output, which can then be stored and processed by 10 Gbit/s measurement nodes.

For measurements on fibre or switched connections running point-to-point protocols (e.g. HDLC), physical access to the network traffic can be gained in three ways:

1. *Port mirroring*: Network devices (typically switches, but also routers) send copies of all packets seen on one or more specific port(s) to a single monitoring port to which a measurement/collection device is connected. The main advantage of this solution is its simplicity and its low cost, since many network devices support this feature out-of-the-box (e.g. Cisco's SPAN-Switch Port ANalyser feature). Furthermore, port mirroring can be remotely administrated and is thus relatively flexible in its configuration. However, there are a number of drawbacks that need to be considered before deciding to use this traffic access method:

   · *Performance*: the mirroring process is an additional burden to the network device's CPU and backplane, which might result in degradation of network performance, especially for heavily utilized devices.

   · *Packet loss*: if the aggregated traffic from the mirrored ports exceeds the capacity of the monitoring ports, packets will be dropped.[2] Even if the capacity of the monitoring port is dimensioned properly, network devices under heavy load might drop packets silently due to the additional CPU burden, which is not of high priority for a network device designed to facilitate traffic transport rather than traffic monitoring.

   · *Packet timing and ordering*: since network devices need to buffer packets from the mirrored links before forwarding them to the monitoring link, timing of packets is affected. As shown in Zhang and Moore [43], port-mirroring on two switches from different vendors introduced significant changes to inter-packet timing and packet-re-ordering, even at very low levels of utilization. These results imply that port-mirroring is likely to introduce bias for all analyzing purposes that include packet inter-arrival time statistics or rely on proper packet arrival order (such as analysis of TCP sequence numbers, etc.).

   · *Omitted packets*: packets with errors in the lower layers of the protocol stack (layers 1 and 2) are usually dropped by network devices and thus not mirrored, which disqualifies port-mirroring for low-layer troubleshooting and debugging purposes.

2. *Port mirroring on dedicated box*: small switches dedicated to mirror link(s) are also called *aggregation TAPs* (Test Access Ports). The main advantage of this solution is increased buffer sizes and a dedicated CPU and backplane, offering some protection against packet loss. However, since this solution requires additional hardware expenses while still not resolving many problems with port mirroring on network devices (packet timing and ordering, omitted packets), it is seldom applied in existing studies dealing with measured network data.

3. *Network TAP*: a network TAP is a device intercepting traffic on a network link, analogous to telephone taps. TAPs are available for copper and optical fiber supporting up to 10Gbit/s. TAPs split the incoming signal into two signals, one signal continuing on the network link and the other signal

[1] up to 64% of CPU usage on two Intel Xeon 5160 dual-core 3.00 GHz CPUs for recording of full packets
[2] Mirroring a full-duplex port requires twice the capacity on the monitoring port (one link in each dir.)

passed on to a measurement/collection device. While copper TAPs use electronic circuits for duplication, fiber TAPs optically split the signal and are thus called *optical splitters*. For duplex links, each direction needs to be tapped individually, and the resulting traffic stream might or might not be merged in the attached measurement device, depending on the specific measurement purpose. Such passive TAPs are non-intrusive and do not interfere with the data or the timing of the data, eliminating most of the drawbacks with port-mirroring. However, in addition to the extra cost, installation of passive TAPs results in a service disruption of the links monitored. Furthermore amplifiers for the optical signal might be required in order to compensate for the power loss due to the optical splitter.

6.2 Data amount

The amount of data carried on modern Internet backbone links makes it non-trivial to record. This will continue to be a challenge in the foreseeable future, since backbone link bandwidths increase in at least the same pace as processing and storage capacities, with 10 Gbit/s links established as state-of-the-art, 40 Gbit/s links already operational and 100 Gbit/s links planned to be introduced in 2010.

6.2.1 Hardware requirements

Increasing link speeds will emphasize hardware requirements of measurement nodes. Some examples of possible bottlenecks within measurement hardware are listed below:

1. *I/O bus*: If high-capacity backbone links operate in full speed, contemporary I/O bus capacities (e.g. 8 Gbit/s theoretical throughput for PCI-X or 16 Gbit/s for 8-lane PCIe 1.x) are hardly sufficient to process data from complete packet header traces. This insufficiency is even more severe when the data needs to pass the bus twice, once to the main memory and another time to secondary storage. Cutting edge PCIe 2.0 or the upcoming PCIe 3.0 featuring 16 or 32 lanes with theoretical throughputs of up to 8 Gbit/s per lane might overcome this bottleneck for current link speeds.

2. *Memory speed*: If the measurement host's main memory is used to buffer traffic before writing it to disk (e.g. to handle bursts in link utilization), it needs to be considered that memory access speeds do not develop in the same pace as link capacities. Modern DDR2-1066 SDRAM DIMMs offer theoretical transfer rates of 68 Gbit/s (8533 MB/s), which would not be sufficiently fast to buffer data from 100 Gbit/s links on full capacity. Only DDR3 SDRAM technology might nominally overcome the 100 Gbit/s border, with I/O clock speeds of up to 800 Mhz (offering transfer rates of 12,800 MB/s or 102.4 Gbit/s). DDR3 DIMMs are expected to penetrate the market throughout the year 2010. However, it is not enough to store data in memory, it eventually also needs to be read out on disks, which doubles the data-rate required. On the other hand, utilization of memory interleaving between multiple memory banks is a common technique to increase memory throughput on many motherboards/chipsets.

3. *Storage speed*: Even if the I/O bus bottleneck could be overcome, the speed of storage array systems would not suffice. Modern storage array network (SAN) solutions offer in the best case 10 Gbit/s rates. Traditional SCSI systems provide nominal throughput rates of around 5 Gbit/s (e.g. Ultra-640 SCSI), and cutting-edge serial buses such as SAS-2.0 (serial attached SCSI) and SATA (serial ATA) reach 6 Gbit/s. Transfer rates for single hard disks range from around 110 MB/s for good disks with 7200 RPM (revolutions per minute) to 170 MB/s sustained transfer rates for the latest 15,600 RPM drives, which can be scaled up by deployment of RAID disk arrays (e.g. RAID-0). These throughput rates could potentially cope with complete packet-level traces of 10 Gbit/s links, but cannot keep up with higher link rates.

4. *Storage capacity*: All these considerations still do not take the required storage capacity into account. Longitudinal measurement campaigns, recording up to several Gigabytes of network data per second, are non-trivial tasks and will eventually be limited by storage capacities.

The discussion provided above shows that recording of complete packet-level traces is strictly bounded by hardware performance, even if it may theoretically be matched with today's hardware. Fortunately, backbone links are typically over-provisioned, and average throughput is far from line-speed. Even though this fact alleviates some technical problems (e.g. storage capacity), measurement nodes still need to be able to absorb temporary traffic bursts. If such traffic amounts cannot be handled, random and uncontrolled discarding of packets will take place, resulting in incomplete, biased datasets, which is highly undesirable with respect to the accuracy of scientific results.

6.2.2 Traffic data reduction techniques

As shown, measurement of complete packet-level traces is technically not always feasible. In the following paragraphs some approaches aiming to reduce data amounts while still preserving relevant information are presented.

1. *Filtering*: If network data are collected with a specific, well defined purpose, traffic filtering is a valid solution to reduce data amounts. Traffic can be filtered according to hosts (IP addresses) or port numbers, which is probably the most common way to filter traffic. But also other arbitrary header fields or even payload signatures can be used as filter criteria. This was already successfully demonstrated by a very early study about Internet traffic characteristics, carried out by Paxson [44]. In this work, only TCP packets with SYN, FIN or RST packets were considered for analysis. Filtering only packets with specified properties can be done in software (e.g. BSD packet filter [4]), which is again limited by processing capabilities, or in hardware (e.g. by FPGAs), which can provide traffic classification and filtering according to a set of rules up to 10 Gbit/s line speeds (e.g. Endace DAG cards [39]).

2. *Sampling*: Another method to reduce data amounts of packet-level traces is packet sampling. Packet sampling can be done systematically, in static intervals (record every Nth packet only) or in random intervals, like proposed by sFlow [45]. Alternatively, also more sophisticated packet sampling approaches have been proposed, such as adaptive packet sampling [46]. A good overview of sampling and filtering techniques for IP packet selections can be found in a recent Internet draft by Zseby et al. [47].

3. *Packet truncation*: A common tradeoff between completeness of packet-level traces and hardware limitations is to truncate recorded packets after a fixed number of bytes. Depending on the chosen byte number, this approach is either not guaranteeing preservation of complete header information or includes potentially privacy-sensitive packet payloads. To address this dilemma, it is common practice to truncate packets in an adaptive fashion, i.e. to record packet headers only. As discussed in Section 4.1, stripping of payload data has also the advantage of addressing privacy concerns. The processing of packets, i.e. the decision of what to keep and where to truncate, can in the best case be done online, especially if hardware support is given. Alternatively, packets can be truncated after a specified packet length of N bytes, and removal of payload is then done during offline processing of the traces.

4. *Flow aggregation*: As discussed in Section 2, a common way to reduce data while still keeping relevant information is to summarize sequences of packets into flows or sessions. The advantage is, that classification of individual packets into flows can be done online, even for high-speed networks due to optimized hardware support of modern measurement equipment. This means that the measurement hosts only need to process and store reduced information in form of flow records, which is no burden even for off-the-shelf servers. Flow records can also be provided by the network infrastructure itself (e.g. by routers), which explains why the most common flow record format NetFlow [**?**] was developed by Cisco. In 2006, the IETF proposed IPFIX [48] as universal flow standard, which is actually derived from NetFlow v9. Even though usage of flow records is already reducing data amounts, various sampling techniques have been proposed for flow collection as well. Flow sampling approaches include random flow sampling (e.g. NetFlow), sample and hold [49] and other advanced sampling techniques, such as in [46, 50, 51].

6.2.3 Archiving of network data

Since measuring Internet traffic is a laborious and expensive task, measurement projects sometimes want to archive not only their analysis results, but also the raw data, such as packet-level traces or flow data. Archiving raw data can be important for several reasons:

1. keeping scientific results reproducible;
2. allowing comparisons between historical and current data;
3. making additional analysis regarding different aspects possible;
4. sharing datasets with the research community, as discussed in Section 7.

Archiving network traces is not always a trivial task, especially for longitudinal, continuous measurement activities. A complete description of different archiving solutions is not within the scope of this paper, but it is recommended to consider risk management such as error handling and redundancy. Data redundancy can be provided by suitable RAID solutions or by periodic backups on tertiary storage such as tape libraries. To further reduce data amounts, compression of traffic traces and flow data for archiving purposes is common practice. Standard compression methods (e.g. Zip) reduce data amounts to 50%, which can be further optimized to 38% as shown in [52]. When network data are archived, it is also crucial to attach descriptive meta-data to datasets, as argued by Pang et al. [29], Paxson [53], and Cleary et al. [54]. Meta-data should include at least descriptions of the measurement and processing routines along with relevant background information about the nature of the stored data, such as network topology, customer breakdown, known network characteristics or uncommon events during the measurement process. To avoid confusion, Pang et al. recommend to associate meta-data to datasets by adding a checksum digest of the trace to the meta-data file.

6.3 Trace sanitization

We define *trace sanitization* as the process of checking and ensuring that Internet data traces are free from logical inconsistencies and are suitable for further analysis. Hence, the goal of trace sanitization is to build confidence in the data collection and preprocessing routines. It is important to take various error sources into account, such as problems with measurement hardware, bugs in processing software and malformed or invalid packet headers, which need to be handled properly by processing and analysis software. Consistency checks can include checksum verification on different protocol levels, analysis of log files from relevant measurement hard- and software and ensuring timestamp consistency. Furthermore, an early basic analysis of traces can reveal unanticipated errors, which might require manual inspection. Statistical properties and traffic decompositions which highly deviate from "normally" observed behavior very often reveal measurement errors (such as garbled packets) or incorrect interpretation of special packets (such as uncommon or malformed protocol headers). Obviously, the results of the trace sanitization process including a documentation of the sanitization procedure should be included into the meta-data of the dataset. An example of a sanitization procedure is described in Section 8.4. Another example of an automated sanitization process is provided by Fraleigh et al. in [55], and a more general discussion about sanitization can be found in Paxson's guidelines for Internet measurement [53].

6.4 Timing issues

Internet measurement has an increasing need for precise and accurate timing, considering that small packets of e.g. 40 bytes traveling back to back on 10 Gbit/s links arrive with as little as 32 nanoseconds (ns) time difference. For each packet a timestamp is attached when recorded, which forms the basic information resource for analysis of time related properties such as throughput, packet-inter-arrival times and delay measurements. Before discussing different timing and synchronization issues occurring in Internet measurement, it is important to define a common terminology about clock characteristics. Next, an overview of timestamp formats will be given, including the important question of when timestamps should be generated during the measurement process. After presenting common types of clocks, this subsection gives a discussion of how accurate timing and clock synchronization can be provided.

6.4.1 Time and clock terminology

First of all it is important to distinguish between a clock's reported time and the true time as defined by national standards, based on the coordinated universal time (UTC [3]). A perfect clock would report true time according to UTC at any given moment. The clock terminology definitions provided below follow Mills' network time protocol (NTP) version 3 standard [56] and the definitions given by Paxson in [57].

- A clock's *resolution*, called *precision* in the NTP specification, is defined by the smallest unit a clock time can be updated, i.e. the resolution is bounded by a clock "tick".
- A clock's *accuracy* tells how well its frequency and time compare with true time.
- The *stability* of a clock is how well it can maintain a constant frequency.
- The *offset* of a clock is the differences between reported time and true time at one particular moment, i.e. the offset is the time difference between two clocks.
- A clock's *skew* is the first derivative of its offset with respect to true time (or another clock's time). In other words, skew is the frequency difference between two clocks.
- A clock' *drift* furthermore is the second derivative of the clock's offset, which means drift is basically the variation in skew.

6.4.2 Generation and format of timestamps

Regardless of how timing information is stored, it is important to understand which moment in time a timestamp is actually referring to. Packets could be timestamped on packet arrival of the first, the last or any arbitrary bit on the link. Software-based packet filters, such as the BSD packet filter [4], commonly timestamp packets after receiving the end of an arriving packet. Furthermore, software solutions often introduce errors and inaccuracies, since arriving packets need to be transported via a bus into the host's main memory, accompanied by an undefined waiting period for a CPU interrupt. Additionally, buffering of packets in the network card can lead to identical timestamps for a number of consecutive packets. These sources of errors are typically not an issue for hardware solutions, such as Endace DAG cards [39]. Another difference is that dedicated measurement hardware generates timestamps on the beginning of packet arrival. If it is for technical reasons not possible to determine the exact start of a packet, timestamps are generated after arrival of the first byte of the data link header (e.g. HDLC), as done by DAG cards for PoS (Packet over SONET) packets [58].

There are also different definitions of how time is represented in timestamps. The traditional Unix timestamp consists of an integer value of 32 bits (later 64 bits) representing seconds since the first of January 1970, the beginning of the Unix epoch. The resolution presented by this timestamp format is therefore one second, which is clearly not enough to meet Internet measurement requirements. PCAP, the trace format of the BSD packet filter, originally supported 64 bit timestamps that indicated the number of seconds and microseconds since the beginning of the Unix epoch. A more precise time stamp format was introduced with NTP [56], representing time in a 64 bit fixed-point format. The first 32 bits represent seconds since first of January 1900, the remaining 32 bits represent fractions of a second. In Endace ERF trace format, a very similar timestamp scheme is used, with the only difference that ERF timestamps count seconds from the start of the Unix epoch (January 1st 1970). These formats can store timestamps with a resolution of 232 pico seconds ($1s/2^{32}$). Currently, the most advanced hardware can actually use 27 bits of the fraction part, providing a resolution of 7.5 ns [59]. Future improvements of clock resolutions will require no modification of timestamp or trace formats but only take advantage of the currently unused bits in the fraction part. Note that the different timestamp formats within different trace formats can have negative effects on trace conversion (Section 2). Converting ERF traces into PCAP traces might imply an undesired reduction of precision from nanosecond to microsecond scale.

6.4.3 Types of clocks

Current commodity computers have typically two clocks. One independent, battery powered *hardware clock* and the *system, or software clock*. The hardware clock is used to keep time when the system

[3] UTC is derived from the average of more than 250 Cesium-clocks situated around the world.

is turned off. Running systems on the other hand typically use the system clock only. The system clock, however, is neither very precise (with resolutions in the millisecond range), nor very stable, with significant skew. In order to provide higher clock accuracy and stability for network measurements, Pasztor and Veitch [60] therefore proposed to exploit the TSC register, a special register which is available on many modern processor types. Their proposed software clock counts CPU cycles based on the TSC register, which offers nanosecond resolution, but above all a highly improved clock stability, with a skew similar to GPS synchronized solutions.

Since tight synchronization is of increasing importance, modern network measurement hardware incorporates special timing systems, such as the DAG universal clock kit (DUCK) [58, 59] in Endace DAG cards. The most advanced DUCK clocks currently run at frequencies of 134 MHz, providing a resolution of 7.5 ns, which is sufficient for packets on 10 Gbit/s links. The DUCK is furthermore capable of adjusting its frequency according to a reference clock which can be connected to the measurement card. Reference clocks (such as a GPS receiver or another DUCK) provide a pulse per second (PPS) signal, which provides accurate synchronization within two clock ticks. For 134 MHz oscillators this consequently means an accuracy of ± 15ns, which can be regarded as very high clock stability.

6.4.4 Clock synchronization

How accurate clocks need to be synchronized when performing Internet measurements depends on the situation and the purpose of the intended analysis. For throughput estimation, microsecond accuracy might be sufficient. On the other hand, some properties, such as delay or jitter on high-speed links, often require higher accuracy. In situations with a single measurement point, instead of accurate timing it might be more important to provide a clock offering sufficient stability. Other situations require tight synchronization with true time, while sometimes it is more important to synchronize two remote clocks, and true time can actually be disregarded. In the following paragraphs, we first present some ways of how to synchronize clocks with each other (where one clock might in fact represent true time). This discussion includes an interesting solution to synchronize measurement hardware located in close proximity, which is especially useful when traces recorded on two unidirectional links need to be merged. Finally, methods allowing correction of timing information retrospectively are presented, which is used to adjust one-way-delay measurements, but can also be applied on passive traffic measurements involving remote measurement locations.

6.4.4.1 Continuous clock synchronization

1. *NTP*: The most common way to synchronize a clock of a computer to a time reference is the network time protocol NTP [56]. NTP is a hierarchical system, with some servers directly attached to a reference clock (e.g. by GPS). Such directly attached servers are called stratum 1 servers. This timing information is then distributed through a tree of NTP servers with increasing stratum numbers after each hop. Depending on the type of the network, the distance to the NTP server and the stratum number of the server, NTP can provide clients with timing accuracy ranging from one millisecond to tens of milliseconds. However, forms of clock skew, drift and jumps despite usage of NTP have been reported by Paxson in [57]. These observations lead to the recommendation to disable NTP synchronization during measurement campaigns, thus providing NTP synchronization only before and after measurement intervals.

2. *GPS*: Since the propagation of timing information over networks obviously limits the accuracy of NTP synchronization, some measurement projects directly attach GPS receivers to their measurement equipment. The global positioning system, GPS, is basically a navigation system based on satellites orbiting the earth. The satellites broadcast timing information of the atomic clocks they carry. GPS receivers, however, can not only be used for positioning, but they can also be used as a time source since highly accurate timing information is received in parallel. GPS receivers can therefore provide clock synchronization within a few hundreds of nanoseconds. Unfortunately, GPS receivers require line of sight to the satellites due to the high frequencies of the signals, which means that GPS antennas normally must be installed outside buildings, ideally on the roof. This can be a severe practical problem, especially for measurement equipment located in data centers in the basement of high buildings.

3. *Cellular networks*: To overcome the practical problems of GPS, it is possible to use signals of cellular telephone networks, such as code division multiple access (CDMA) as synchronization source for measurement nodes (e.g. provided by [61]). Base stations of cellular networks are all equipped with GPS receivers to retrieve timing information. This information is then broadcasted as a control signal within the network. Since base stations operate on lower frequencies, it is possible to use these base stations as timing sources even inside buildings. The accuracy provided by CDMA time receivers is very close to GPS standards. However, due to the unknown distance to the base station, clocks synchronized by CDMA will have an unknown offset from UTC. Furthermore, the offset is not guaranteed to be constant, since receivers in cellular networks can switch between base stations for various reasons.

4. *SDH protocol*: A recently proposed approach distributes time from an UTC node using existing backbone communication networks, such as OC192 links. This system yields an accuracy of a few nanoseconds, which is done by utilizing the data packages already transmitted in the system [62]. To our knowledge, this novel approach has not been used in Internet measurement yet, but it might be an interesting alternative for upcoming measurement projects.

5. *Daisy-chaining of timestamping clock*: Endace DAG cards offer an additional solution for clock synchronization, which is very attractive for measurement hosts located in close proximity. The DUCK, a clock kit on every DAG cards, offers also output of PPS signals [59]. This feature can be used to chain DAG cards together by simple local cabling in order to keep them tightly synchronized. If no external reference clock is available, at least accurate and consistent timestamping between the connected DAG cards is provided. This approach is often used when two links in opposing directions are measured with two separate measurement hosts, since it allows merging of the traces into one bidirectional trace. In this case, synchronization between the two clocks is of main importance, and accuracy with respect to true time (UTC) is no major concern.

6.4.4.2 Retrospective time correction

In some cases (e.g. for large geographical distances), traffic traces timestamped by different clocks need to be compared. Even if clock synchronization by NTP or GPS is provided, forms of clock skew, drift and jumps cannot be ruled out [57]. To compensate for these errors, retrospective time correction algorithms have been proposed. These algorithms have been designed to remove clock offset and skew from one-way delay (OWD) measurements. For distributed passive traffic measurements a set of passive OWD measurements can be obtained given that sufficient (uniquely identifiable) packets traverse both measurement locations. In this case, the correction algorithms can also be applied on passive packet traces collected at different locations.

The observed OWD (OOWD) for the ith packet can be calculated as

$$OOWD(i) = t_r(i) - t_s(i) \tag{1}$$

where t_r and t_s are the timestamps of the ith packet at receiver and sender respectively. Given a relative clock offset δ between the receiver and sender clock, the actual OWD can then be derived by:

$$OWD(i) = OOWD(i) - \delta(i) \tag{2}$$

Early approaches assumed zero clock drift (i.e. constant clock skew) and no instantaneous clock adjustments in order to estimated clock skew. This means that a series of OWD(i) measurements would indicate a trend, following steady increasing or decreasing $\delta(i)$, depending on the sign of the clock skew according to the reference clock (typically the receivers clock). Moon et al. [63] proposed a *linear programming-based algorithm* in order to estimate the slope α of the resulting trend starting at an initial offset β, i.e. $\delta(1)$.

Newer approaches also try to take clock dynamics into account by partitioning the measurements into segments representing periods with constant skew between clock jumps or frequency adjustments. However, even these newer approaches assume zero drift between the two clocks. Zhang et al. [64] proposed a set of algorithms based on the computation of *convex hulls* in order to remove skews. A devide-and-conquer approach is used to identify clock resets (i.e. jumps) and a marching algorithm

should identify epochs of frequency adjustments, between which the relative clock skew is estimated and removed. However, Zhang's approach has some limitations, such as limitations of how often and frequent clock resets can occur.

Wang et al. [65] tried to generalize previous approaches by converting the clock dynamics detection to a *time series segmentation* problem. The resulting clustering based OTDTS algorithm (Optimized Top-Down Time series Segmentation) segments delay time series at the points at which clock resets or adjustments occur. For each segment, clock skew can then be estimated and removed either by the linear programming-based algorithm as in Moon et al. or the convex hull approach as proposed by Zhang et al.

A fuzzy-based approach for estimation and removal of clock skew and reset has been proposed by Lin et al. [66], which is claimed to be more accurate and robust than Zhang's convex-hull approach. The authors combine the fuzzy clustering analysis [67] with the linear programming-based or the convex-hull approach, where the fuzzy analysis is used to distinguish between clock resets and temporary delay variations such as traffic congestions.

Khlifi and Gregoire [68] tried to further reduce the complexity of previous skew estimation approaches such as linear programming and convex-hull. Two techniques for offline skew removal are proposed. The *average technique*, which reduces the complexity of previous algorithms from O(N) to O(1) by calculating the average of the delay differences between consecutive packets. The *direct skew removal technique* remains at O(N) complexity while increasing the accuracy by iteratively evaluating the set of possible skews until an optimal value is reached. Furthermore, two online techniques for skew removal are proposed, namely the *sliding window algorithm* which tracks the skew by continual evaluation of variations in the minimum OOWD and a *combined algorithm*, combining the sliding window and convex-hull approaches.

7 Data sharing

The discussions about all the legal, operational and technical difficulties involved in conducting Internet measurement clearly show that proper network traces are the result of a laborious and costly process. This explains why currently only few researchers and research groups have the possibilities to collect Internet backbone data, which makes proper traces a scarce resource. Therefore, the Internet measurement community has repeatedly been encouraged to share their valuable datasets and make them publicly available [69, 53, 33], given that sharing of network data are legally permitted (see Section 3). Sharing network data are not only a service to the community, but it is also an important factor related to credibility of research results. Generally, sharing and archiving of data are fundamental to scientific progress and help to improve scope and quality of future research.

Sharing network traces adds reliability to research, since it makes results reproducible by the public, which allows verification and in the best case confirmation of earlier results. This should be best practice in research, encouraging fruitful research dialogs and discussions within the research community. Furthermore, releasing measurement data makes it possible to compare competing methods on identical datasets, allowing fair and unbiased comparison of novel methodologies. Publishing of network data also gives the additional benefit of providing the original data owners with supplementary information about their data, yielding a better and more complete understanding of the data. Finally, in order to get a representative view of the Internet, diverse data at different locations and times needs to be collected and shared within the research community. In a note on issues and etiquette concerning use of shared measurement data [33], Allman and Paxson discuss the above-mentioned benefits of data availability, including ethic and privacy considerations, as discussed here in Section 4.

Before data can actually be shared, researchers need to be made aware of existing and available datasets. A system for sharing Internet measurements was proposed by Allmann in 2002 [70]. This system was inspiration for CAIDA to finally implement the Internet measurement data catalog DatCat [71], which allows publication of meta-data about network datasets. The goal of this project was to provide the research community with a central database, providing searchable descriptions of existing datasets.

Actual sharing of data, however, is problematic due to the mentioned uncertain legal situation and ethical considerations. Even if traces are desensitized by technological means (e.g. by payload removal and anonymization), additional sharing policies are required in order to safeguard possible technological shortcomings such as trace de-anonymization (see Section 4.2.2). Kenneally and Claffy therefore try to

facilitate protected data sharing by proactively implementing management of privacy risks in the Privacy-Sensitive Sharing framework PS2 [17]. PS2 is based on an hybrid model relying on a policy framework applying privacy principles together with a technology framework implementing and enforcing privacy obligations. So far, the authors have only outlined the PS2 framework without focusing on a particular implementation of a data sharing tool.

An alternative approach to data sharing was suggested by Mogul in a presentation in 2002 [72]. He proposes a "move code to the data" solution, where analysis programs are sent to the data owners (e.g. network operators) and executed on-site. In this scenario, only results would be shared, but not the network data itself. This is an interesting approach, but it highly depends on the will of the involved parties to cooperate.

As a solution to the privacy/utility tradeoff in data sharing, Mirkovic [73] proposed a privacy-safe sharing framework based on secure queries. Instead of sharing (copies of) raw traces, data access is re-directed through an online interface providing a query language, allowing customized sets of queries to be run on the data and returning de-sensitized, aggregated information fitting the specific research goals. Individual privacy policies can thus be enforced by the query language interpreter.

Parate and Miklau [74] very recently proposed a sharing framework in which trace owners can match an anonymizing transformation of communication data with the requirements of analysts. The framework should so enable formal reasoning about the impact of anonymization operations on trace utility and privacy.

CASFI (Collect, Analyze, and Share for the Future Internet) is currently working on the CASFI Data Sharing Platform [75], a framework that helps to share not only data, but also the data management platform to facilitate better collaboration between multiple research groups. The platform should help to manage local data, and at the same time provide an interface to remote data, in order to get a consistent overview of relevant data without visiting different web interfaces.

8 Experiences from the MonNet project

This section provides a description and lessons learned from a project for passive Internet traffic monitoring and analysis conducted at Chalmers University of Technology: the MonNet project. The goal of the project is to provide a better understanding of Internet traffic characteristics based on empirical data, i.e. passive measurements on backbone links.

8.1 Legal and ethical approval

In summer 2004, MonNet, as a project regarding Internet and traffic measurements and analysis, was proposed to the SUNET board. In order for the project to be granted, the SUNET board required permission from the "central Swedish committee for vetting ethics of research involving humans" (*Etikprövningsnämnden, EPN*), which is among other things responsible for vetting research that involves dealing with sensitive information about people or personal information. Ethical vetting in this committee is carried out in six regional boards. After elaborate discussions about the de-sensitization process of the traces, the regional ethics committee permitted the MonNet measurements to take place. Traffic monitoring was granted under the conditions that user payload is removed and IP addresses are anonymized, e.g. with prefix-preserving Crypto-PAn. We consider the provided permission from the research ethics committee as an appropriate safeguard, as requested for measurement of Internet traffic for scientific purposes by current EU directives (see Section 3.1).

Lessons learned:

1. During the vetting process, it turned out that the committee had little understanding of the technical background and implications. This resulted in a policy suggested by the MonNet project itself which, after some amendments, was approved by the vetting committee. Researchers therefore need to be aware of how and on which level of detail policies for de-sensitization and sharing are formulated in order not to hinder sound scientific research while still respecting privacy of individuals.

2. Obtaining legal approval can introduce a long and unpredictable time delay, which needs to be considered in the project planning.

8.2 Operational considerations

Operational considerations include choice of measurement location and access to measurement premises:

8.2.1 Measurement location

Before actual measurements could be started, a measurement location needed to be chosen. The MonNet project initially (from 2005 until 2007) recorded traffic on GigaSUNET Internet backbone links. Data were collected on the outside point of an OC192 ring, which was the primary link from the region of Gothenburg to the main Internet outside Sweden. The link carried traffic from major universities, large student residential networks and a regional access point exchanging traffic with local ISPs. This choice was in the first place made to achieve traces with a high level of traffic aggregation, since the link measured carries data transferred between a regional network and the main Internet.

The former ring architecture has during 2007 been upgraded to OptoSUNET, a star structure over leased fiber. All SUNET customers are since then redundantly connected to a central Internet access point in Stockholm. Besides some local exchange traffic, the traffic routed to international commodity Internet is carried on two links (40 Gbit/s and 10 Gbit/s) between SUNET and a Tier-1 provider. Since 40 Gbit/s measurement equipment was economically impossible (it would essentially require measurement equipment for 4 x 10 Gbit/s links), the measurement infrastructure was moved to the 10 Gbit/s link with the highest possible level of traffic aggregation: the 10 Gbit/s link between SUNET and NorduNet, located in Stockholm. According to SNMP statistics, this link carries 50% of all inbound but only 15% of the outbound traffic volume.[4] In July 2009 an additional 10 Gbit/s link was installed in parallel with the existing one in order to keep up with the increasing traffic volumes.

8.2.2 Access to premises

The initial measurement location had the additional feature of being located in the same city as the research group, at the Chalmers University of Technology in central Gothenburg. This feature was of great advantage for very practical reasons:

- installation of optical splitters and operation of specialized measurement cards on 10 Gbit/s speeds could not be tested beforehand in a lab environment, which required some adjustments on site as reaction on early experiences.
- even tested commodity PCs used as measurement nodes required additional physical visits at the measurement location due to unexpected early hardware defects such as harddisk crashes and RAID controller problems, which are most common in early and very late stages of hardware lifecycles (following the bathtub-curve).

Even if located in the same city, physical access to the actual measurement location, situated in secure premises of an external network operator, was not entirely straight-forward to obtain and involved inconvenient administrative overhead and idle times. Limited access possibilities to operational network facilities and equipment is common for many network research projects and should be taken into account when negotiating rules and policies with the network operator enabling the measurements. To prevent some of the physical visits, it is useful to equip nodes with remote management hardware capable of hardware resets and access to the system console. Unfortunately, the remote management cards recommended by the supplier of MonNet's measurement nodes turned out to be unstable and unreliable (i.e. useless), which highlights the importance of using as much well-established and tested hardware as possible. As a consequence, SSH access, which was granted only to specified hosts inside Chalmers University, was the only way to remotely maintain and operate the measurement nodes.

Lessons learned:

1. Good contacts with the network's operators and well defined access procedures alleviate installation and configuration of measurement equipment.

[4] 85% of the outbound traffic is routed via the 40 Gbit/s link.

2. Proven remote management possibilities should be exploited.

3. Measurement locations should in the first place be chosen in order to provide data supporting the specific research purpose. If possible, it is of advantage to choose geographically close locations, which is especially true in early project phases.

4. Unforeseen delays by external parties (ethic committee, operators, hardware suppliers) require sufficient slack times and parallel work, especially in early project phases.

5. In face of frequent changes to network topologies and technologies, measurement hardware supporting various link-layer technologies (e.g. PoS HDLC and GbE) and wavelengths is preferable since it can be reused without additional costs.

8.3 Technical solution

The measurement nodes applied have been designed to meet the anticipated requirements of packet-header measurements on PoS OC192 links. During the planning phase, related measurement projects such as NLANR PMA's OC48MON [76] and Sprint's IPMON [55] provided valuable inspiration. The described technical solution is based on state-of-the-art hardware available during the design phase in 2004.

8.3.1 Traffic access

Optical splitters on two OC-192 links, one for each direction, are used to capture PoS HDLC traffic. Since the signal strength was quite high, splitters with a 90/10 ratio turned out to be sufficient for the sensitivity of the measurement cards, while not requiring any additional signal amplifiers on the production links. Each optical splitter, tapping either the inbound or outbound OC912 link, is attached to an Endace DAG6.2SE card sitting in one of the measurement nodes via a PCI-X 133 MHz 64-bit PCI interface. The DAG cards have been configured with a buffer reserved from the node's main memory in order to deal with burst of high traffic load. For Linux systems, the buffer size needs to lie between 128 MB and 890 MB. Endace recommends a buffer size of at least 32MB for OC-12 links, thus we conservatively chose a large buffer of 512 MB for OC-192 links, which did not result in any packet loss due to insufficient buffer space in any of our measurements.[5] DAG6.2SE cards can capture 10 Gbit/s on links with optical wavelengths between 1300 and 1600 nm with STM-64c, 10GbE WAN and 10GbE LAN encapsulations. Packets are timestamped with a resolution of 15 ns.

8.3.2 Collection hardware

The two measurement nodes are designed and configured identically. A schematic block diagram of the relevant components is shown in Fig. 1. A measurement node consists of two AMD Opteron 64-bit processors with 2 GHz clock frequency and a total of 2 GB of main memory, 1 GB per CPU as two interleaved 512 MB DDR-400 SDRAM DIMMs. The Tyan K8SR motherboard is equipped with an AMD-8131 PCI-X Tunnel chipset connecting the processing units with I/O devices on PCI-X slots. The DAG6.2SE card is the only device attached to the 133 MHz 64-bit PCI-X slot. On slot 2, supporting 100 MHz, six SCSI disks are connected to a dual-channel Ultra-320 SCSI controller. The SCSI disks are configured to operate in RAID0 (striping), and thereby add up to about 411 GB of cumulated disk-space for preliminary storage of collected network traces. The 6 Maxtor Atlas SCSI disks reach a sustained data rate of between 40 and 72 MB/s, depending on the cylinder location. A series of tests with sequential writes on the RAID0 system resulted in an average data rate of about 410 MB/s (3.3 Gbit/s). Furthermore, a mirrored RAID-1 disk containing the operating system is connected to the IDE controller (not visualized in Fig. 1).

As evident in Fig. 1, the bottleneck of this configuration is the storage system, with about 3.3 Gbit/s throughput. But also the nominal throughput of the SCSI interface (5.2 Gbit/s) and the PCI-X buses (8.5 and 6.4 Gbit/s, respectively) are not sufficient to collect full packet traces in full line speed on 10 Gbit/s networks. Also note that the buses above the PCI-X 8131 tunnel in the figure are traversed twice

[5] Dropped packets due to insufficient buffer space, PCI bus limitations or losses between the DAG card and the memory are reported by DAG cards

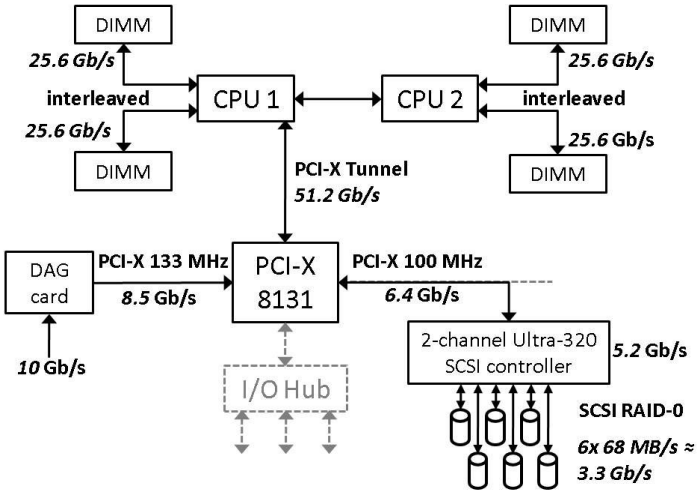

Figure 1: Block diagram of measurement nodes

during measurements (from the DAG Card into Memory, and then back to the storage), which, however, does not pose a bottleneck in the present configuration.

8.3.3 Time synchronization

During measurements, the two DAG cards have been synchronized with each other using Endace's DUCK Time Synchronization [58, 59] with no external reference time. Before and after measurements, the DAG cards were synchronized to true time (UTC) using a pool of three stratum 1 NTP servers. NTP synchronization was disabled during the measurements, since forms of clocks skew, drift and jumps despite usage of NTP are problematic as described earlier. DUCK, however, can provide an accurate and consistent timestamping between the connected DAG cards ranging between ± 30ns according to Endace [59], even though their time might not be accurate with respect to true time. The tight synchronization between the measurements of opposing traffic directions allows simple merging of the unidirectional data into bidirectional traces.

8.3.4 Processing and archiving platform

After data collection and a pre-processing procedures on the measurement nodes, the resulting traces have been transferred via a Gigabit-Ethernet interface and a 2.5 Gbit/s Internet connection to the storage and processing server located in a secured server room at Chalmers University. The processing platform is attached to an external SCSI array box with a RAID5 configuration, providing 2 TB of storage. 2 TB can store around 35 hours of compressed, bidirectional packet header data collected on the current measurement location in OptoSUNET. Since this is not sufficient for longitudinal measurement campaigns, an archiving solution was required. Due to a tight economic situation, this was solved by acquisition of relatively cheap 1 TB SATA disks, which have been temporary attached via USB. After archiving the data, the disks have been placed offline in a safe when not in use. With the rapidly decreasing storage costs during recent years, it was possible to install an additional 3 TB NAS (network array storage system) with RAID5 configuration acting as online (though slow) archiving system.

Lessons learned:

1. Passive TAPs (optical splitters) in combination with specialized measurement cards is the only way to ensure lossless and precise traffic measurement on high-speed links, which is required in many research situations.

2. Since measurement cards are disproportionally expensive compared to commodity equipment, it is worth it to invest in one measurement node per link (instead of multiple measurement cards in one node) with high quality state-of-the-art hardware components.

3. Measurement nodes need to be designed carefully - performance of each component needs to be considered in order to identify possible bottlenecks.

4. Time synchronization by daisy chaining the DAG cards worked very well and was straight forward, avoiding a lot of timing-related problems (such as need for retrospective time corrections) when analyzing or merging the unidirectional traces.

5. Archiving of network traces should be considered from the beginning, since it inevitable needs to be solved and thus needs to be part of the technical and economical planning.

8.4 Trace pre-processing

Pre-processing of traffic traces including reduction, de-sensitization and sanitization, is carried out on the measurement nodes during and immediately after the collection.

8.4.1 Traffic reduction

The DAG cards have been configured to capture the first 120 bytes of each PoS frame to ensure that all link-, network-, and transport-headers are preserved. The remaining payload fractions have been removed later during the pre-processing of the traces. The average packet size on the links lies around 700 bytes, which means a maximal throughput of around 1.8 million frames per second on a 10 Gbit/s link. 44% of all frames are smaller than 120 bytes and thus not truncated by the DAG card. As a result, the average size of packets that need to be stored on disk after truncation is 88 bytes.This means that even at maximum link utilization of 10 Gbit/s, only about 160 MByte/s need to be transferred to disk with this setup and packet distribution. However, due to heavy over-provisioning of the links measured, in reality the nodes rarely needed to store more than 35 MByte/s (280 Mbit/s) on disk during the MonNet measurement campaigns. Occasional traffic spikes can of course reach much higher throughput values, but these short spikes could successfully be buffered in the reserved main memory (512 MB).

8.4.2 Trace de-sensitization

After storing truncated packets on disks, the traces have been de-sensitized in offline fashion on the measurement nodes, since online pre-processing in real time is unfeasible due to computational speed. De-sensitization has therefore been carried out by batch jobs immediately after collection in order to minimize the storage time of unprocessed and privacy-sensitive network traces.

As a first step in the de-sensitization process, the remaining payload beyond transport layer was removed using CAIDA's *CoralReef* [12] *crl_to_dag* utility. During this processing step, CoralReef also anonymized IP addresses in the remaining headers using the prefix-preserving *Crypto-PAn* [30]. A single, unique encryption-key was used throughout all MonNet measurement campaigns in order to allow tracing of specific IP addresses during the whole time period and for all measurements. This encryption key is kept secure and used for anonymization on the measurement nodes only.

8.4.3 Trace sanitization

Trace sanitization refers to the process of checking and ensuring that the collected traces are free from logical inconsistencies and are suitable for further analysis. This was done by using available tools such as the *dagtools* provided by Endace, accompanied by own tools for additional consistency checks. These checks have been applied before and after each de-sensitization process. Resulting statistical figures such as byte and record numbers have been compared between consecutive passes of the sanitization procedures. In the common cases, when no inconsistencies or errors have been detected, the original, unprocessed traces have been deleted upon completion of the pre-processing procedures, and only de-sensitized and sanitized versions of the traces have been kept. If major errors such as packet loss have been detected, the pre-processing procedure has been stopped, the particular traces on both measurement nodes (for both directions) have been deleted and an immediate new collection has been scheduled automatically on both nodes. Detection of minor errors, such as single checksum inconsistencies, has been documented in the meta-data. For errors of unknown severity further steps have been postponed, requesting manual inspection. The sanitization procedure included the following checks:

Major errors (discarding of traces)

- Are timestamps strictly monotonically increasing?
- Are timestamps in a reasonable time window?
- Are consecutive timestamps yielding feasible inter-arrival times according to line-speed and packet sizes?
- Are frames received continuously? (Packet arrival rates of zero packets/s should not happen on productive backbone links.)
- Are there any occurrences of identical IP headers within consecutive frames?
- Are all recorded frames of known type (i.e. POS with HDLC framing)?
- Is there an unreasonably high number of non-IP (v4 and v6) packets (which indicates garbled data)?
- Has the DAG reported loss of records during transfer to main memory (I/O bus limits)?
- Has the DAG reported packet loss or truncation due to insufficient buffer space?
- Are record counts before and after de-sensitization matching (i.e. no packets have been discarded)?

Minor errors (report in meta-data)

- Are there any IP header checksum errors?
- Have there been any receiver errors (i.e. link errors, such as incorrect light levels on the fiber and HDLC FCS (CRC) errors)?

Unknown severity (manual inspection)

- Did the system log show any error messages during the measurements (e.g. by the measurement card or storage system)?
- Have there been any other internal errors reported in-line by the DAG card?

The sanitization process revealed some traces that had to be discarded due to garbled data or packet arrival rates of zero after a certain time, particularly on one measurement node. We suspect that this particular DAG card sometimes looses framing due to a hardware failure. Furthermore, infrequently the DAG cards discard single frames due to receiver errors, typically HDLC CRC errors. Some frames can be reported as corrupted by the sanitization process due to IP checksum errors. Since the HDLC CRC was shown to be correct, there are cases when the IP checksum and CRC disagree [77]. Another explanation could be checksum errors already introduced by the sender, coupled with routers on the path ignoring the IP checksum in their validation of incoming IP packets and only performing incremental updates [78]. Since such missing or corrupted packets occur very rarely, the traces have still been used for analysis, but missing packets and IP checksum errors have been documented in the attached meta-data file.

Lessons learned:

1. Over-provisioning and packet truncation (as often required for privacy reasons anyhow) reduce hardware requirements and alleviate possible bottlenecks.

2. Thorough trace sanitization after collection and de-sensitization is important in order to avoid waste of computational resources and storage space. Furthermore it is imperative to ensure sound and unbiased scientific results during traffic analysis.

3. Collection circumstances (hardware, software, link, time) and pre-processing steps should be documented in meta-data and attached to the traces throughout the trace lifetime (from collection to archiving of the data).

4. Even if traffic data is sanitized, syntactical problems with individual packets need to be anticipated. This means that pre-processing and analysis tools need to be robustly designed in order to be able to handle all sorts of unknown protocols and packet header anomalies [27].

8.5 Data sharing policy

During the start-up phase, when the MonNet project was planned, vetted and later granted, we missed to establish a clear data sharing policy. After collecting the first traces and publishing results in scientific conferences and journals, other researchers identified our project as a possible resource of recent Internet data and asked for access to MonNet traffic traces. In absence of policies agreed upon by the network provider and the vetting committee, a "move code to data" approach was chosen, in which MonNet project members act as proxy (one level of indirection) between external researchers and the traffic traces.

Lessons learned: Data sharing is an essential part of scientific work, which needs to be explicitly considered already in early project phases. A technological and policy framework that might help future projects to implement secure data sharing is currently being suggested by Kenneally and Claffy [17].

8.6 Traffic analysis and scientific results

So far, only the measurement processes including data pre-processing have been discussed. In this Section, the analysis approaches used to extract scientific results are outlined briefly in order to indicate the applicability and value of Internet measurements [79].

Packet-level analysis: In one of our early studies [80], Internet backbone traffic has been analyzed in order to extract cumulated statistical data into a database. The main challenge in this analysis program was to provide sufficient robustness, i.e. being able to deal with any possible kind of header inconsistency and anomaly. The resulting database consists of tables for specifically interesting features, such as IP header length, IP packet length, TCP options and different kinds of anomalous behavior, which could be analyzed conveniently with the help of SQL queries. A later follow up study [27] provided a more systematic listing of packet header anomalies in order to discuss potential security problems within Internet packet headers.

Flow-level analysis: In order to be able to conduct a detailed connection level analysis, the tightly synchronized unidirectional traces have been merged according to their timestamps. In the resulting bidirectional traces directional information for each frame was preserved in a special bit of the ERF trace format. As a next step, an analysis program collected per-flow information of the packet-level traces. Packet streams have then been summarized in flows by using a hash-table structure in memory. The gathered per-flow information includes packet and data counts for both directions, start- and end times, TCP flags and counters for erroneous packet headers and multiple occurrences of special flags like RST or FIN. This information was inserted into one database table for each transport protocol, each row representing a summary of exactly one flow or connection. The resulting flow database was used to study directional differences [9], increasing portions of UDP traffic [81] and routing symmetry [82] in Internet backbone traffic.

Traffic classification: To get a better understanding of traffic composition, different traffic classification methods have been studied [83]. A first approach to classify traffic on application level was done based on a set of heuristics regarding connection patterns of individual endpoints in the Internet [10]. The resulting classified flow table then allowed us to analyze and compare flow and connection characteristics among traffic of different network applications [11]. Recently, a classification approach based on statistical protocol properties has been suggested [84] and is currently further investigated and evaluated.

Lessons learned: Analysis of packet pevel-data often produces extensive result-sets, even if processed and aggregated. While many researchers and available analysis tools handle and process results-sets on file-level, our experience shows that it is advisable to exploit database systems (e.g. MySQL), since databases are designed to handle large data amounts and facilitate data-mining.

9 Summary and conclusions

The development of the Internet has without doubt not yet come to an end. In the next years, we have to expect a continuing growth in numbers of users and amounts of traffic. Traffic will exhibit an even higher diversity, with the Internet becoming a more and more unified backbone for all forms of communication and content (e.g. VoIP, IPTV, etc.). As a consequence, network bandwidths will continue to increase with at least the same pace as computer processing and storage capacities. However, the ability to keep up with link speeds will not be the only challenge for Internet measurement. There are a number of technical and commercial applications which could directly benefit from results of Internet

measurement and analysis, including network design and provisioning, improvement of network protocols and infrastructure but also network performance and accounting. Analysis of actual Internet traffic is also crucial input for network modeling and further development of network services. The Internet community will therefore have an increasing need for methods and means to collect, analyze, interpret and model Internet traffic.

The success and popularity of the Internet has unfortunately also lead to a rapid increase in all forms of misuse and unsocial, malicious activities - a trend, which is very likely to exacerbate as the importance of the Internet continues to grow. Network security measures, such as intrusion detection and prevention, are depending on profound understanding of traffic properties and have to rely on fast and reliable analysis methods of network anomalies and detection of vulnerabilities. Therefore research on modern, real-life datasets is vital for network security research in order to remain proactive.

Research on technologies and methods to monitor and measure Internet traffic are also of increasing legal relevance. With the data retention directive of the European Union [15], providers in member states will soon be required to retain connection data for periods of up to two years. While this directive could be postponed until March 2009, governments and operators currently need to establish the possibilities to execute the directive. This type of regulation obviously requires adequate technical solutions and know-how - which can both be provided by past, but also upcoming achievements of the Internet measurement and analysis community.

Analysis of Internet traffic is for obvious reasons heavily depending on the quality of existing network traces. It is therefore crucial to provide continuous possibilities to monitor and measure Internet traffic on as many sites as possible while at the same time maintaining respect for moral and ethical constraints. Acquiring network traces on backbone links, however, is a non-trivial task. Our experience shows that many problems can be avoided by careful and anticipatory planning. To facilitate the setting-up of future measurement projects, this paper is intended to serve as a guide for practical issues of Internet measurement based on lessons learned during the MonNet project. The paper addresses the main challenges of passive, large-scale measurements, including legal, ethical, technical and operational aspects. Furthermore, a detailed overview of the research field is given by describing different design choices and state-of-the-art solutions. This paper should provide researchers and practitioners with useful guidelines to setting up future monitoring infrastructures - which will in turn help to improve results from traffic analysis and therefore contribute to a better and more detailed understanding of how the Internet functions.

Acknowledgements

This work was supported by SUNET, the Swedish University Computer Network.

References

[1] R. Nelson, D. Lawson, P. Lorier, Analysis of long duration traces, SIGCOMM Computer Communication Review 35 (1) (2005) 45–52.

[2] A. Householder, K. Houle, C. Dougherty, Computer attack trends challenge internet security, Computer 35 (4) (2002) 5–7.

[3] RIPE NCC, YouTube Hijacking: A RIPE NCC RIS case study, `http://www.ripe.net/news/study-youtube-hijacking.html` (accessed 2009-10-27).

[4] S. McCanne, V. Jacobson, The BSD packet filter: A new architecture for user-level packet capture, in: USENIX Winter, 1993, pp. 259–270.

[5] S. Ubik, P. Zejdl, Passive monitoring of 10 gb/s lines with pc hardware, in: TNC '08: Terena Networking Conference, Bruges, BE, 2008.

[6] R. Braden, Requirements for Internet Hosts - Communication Layers, RFC 1122 (Standard) (1989).

[7] J. Case, M. Fedor, M. Schoffstall, J. Davin, Simple Network Management Protocol (SNMP), RFC 1157 (Historic) (1990).

[8] B. Claise, Cisco Systems NetFlow Services Export Version 9, RFC 3954 (Informational) (2004).

[9] W. John, S. Tafvelin, Differences between in- and outbound internet backbone traffic, in: TNC '07: Terena Networking Conference, 2007.

[10] W. John, S. Tafvelin, Heuristics to classify internet backbone traffic based on connection patterns, in: ICOIN '08: International Conference on Information Networking, 2008, pp. 1–5.

[11] W. John, S. Tafvelin, T. Olovsson, Trends and differences in connection-behavior within classes of internet backbone traffic, in: PAM '08: Passive and Active Network Measurement Conference, 2008, pp. 192–201.

[12] K. Keys, D. Moore, R. Koga, E. Lagache, M. Tesch, k claffy, The architecture of CoralReef: an Internet traffic monitoring software suite, in: A workshop on Passive and Active Measurements, PAM '01, 2001.

[13] Directive 95/46/ec of the european parilament and of the council, http://ec.europa.eu/justice_home/fsj/privacy/docs/95-46-ce/dir1995-46_part1_en.pdf (1995).

[14] Directive 2002/58/ec of the european parilament and of the council, http://eur-lex.europa.eu/LexUriServ/site/en/oj/2002/l_201/l_20120020731en00370047.pdf (2002).

[15] Directive 2006/24/ec of the european parilament and of the council, http://eur-lex.europa.eu/LexUriServ/site/en/oj/2006/l_105/l_10520060413en00540063.pdf (2006).

[16] AK-Vorrat, Overview of national data retention policies, https://wiki.vorratsdatenspeicherung.de/Transposition (accessed 2009-10-27).

[17] E. E. Kenneally, kc claffy, An internet data sharing framework for balancing privacy and utility, in: Proceedings of Engaging Data: First International Forum on the Application and Management of Personal Electronic Information, 2009.

[18] D. C. Sicker, P. Ohm, D. Grunwald, Legal issues surrounding monitoring during network research, in: IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, 2007, pp. 141–148.

[19] 18 united states code §2511, http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00002511----000-.html.

[20] 18 united states code §3127, http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00003127----000-.html.

[21] 18 united states code §2701, http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00002701----000-.html.

[22] 18 united states code §2702, http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00002702----000-.html.

[23] 18 united states code §2703, http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00002703----000-.html.

[24] kc claffy, Ten things lawyers should know about internet research, Tech. rep., CAIDA,SDSC,UCSD, http://www.caida.org/publications/papers/2008/lawyers_top_ten/lawyers_top_ten.pdf (accessed 2009-07-03).

[25] kc claffy, Internet as emerging critical infrastructure: what needs to be measured?, in: JCC'08: Chilean Computing Week, 2008, http://www.caida.org/publications/presentations/2008/uchile/uchile.pdf (accessed 2009-10-27).

[26] T. Karagiannis, A. Broido, N. Brownlee, K. Claffy, M. Faloutsos, Is p2p dying or just hiding?, in: GLOBECOM '04. IEEE Global Telecommunications Conference, Vol. Vol.3, Dallas, TX, USA, 2004, pp. 1532 – 8.

[27] W. John, T. Olovsson, Detection of malicious traffic on backbone links via packet header analysis, Campus Wide Information Systems 25 (5) (2008) 342 – 358.

[28] S. Coull, C. Wright, F. Monrose, M. Collins, M. Reiter, Playing devil's advocate: Inferring sensitive information from anonymized network traces, in: Proceedings of the Network and Distributed Systems Security Symposium, San Diego, CA, USA, 2007.

[29] R. Pang, M. Allman, V. Paxson, J. Lee, The devil and packet trace anonymization, SIGCOMM Comput. Commun. Rev. 36 (1) (2006) 29–38.

[30] J. Xu, J. Fan, M. H. Ammar, S. B. Moon, Prefix-preserving ip address anonymization: Measurement-based security evaluation and a new cryptography-based scheme, in: ICNP '02: Proceedings of the 10th IEEE International Conference on Network Protocols, Washington, USA, 2002, pp. 280–289.

[31] T. Ylonen, Thoughts on how to mount an attack on tcpdpriv's -a50 option, Web White Paper, http://ita.ee.lbl.gov/html/contrib/attack50/attack50.html (accessed 2009-07-03).

[32] T. Kohno, A. Broido, K. C. Claffy, Remote physical device fingerprinting, IEEE Transactions on Dependable and Secure Computing 2 (2) (2005) 93–108.

[33] M. Allman, V. Paxson, Issues and etiquette concerning use of shared measurement data, in: IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, 2007, pp. 135–140.

[34] Acm workshop on network data anonymization, http://www.ics.forth.gr/~antonat/nda08.html (accessed 2009-07-03).

[35] G. Minshall, Tcpdpriv: Program for eliminating confidential information from traces, http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html (accessed 2009-07-03).

[36] A. Slagell, J. Wang, W. Yurcik, Network log anonymization: Application of crypto-pan to cisco netflows, in: SKM '04: Proceedings of Workshop on Secure Knowledge Management, Buffalo, NY, USA, 2004.

[37] R. Ramaswamy, N. Weng, T. Wolf, An ixa-based network measurement node, in: Proceedings of Intel IXA University Summit, Hudson, MA, USA, 2004.

[38] T. Brekne, A. Årnes, Circumventing ip-address pseudonymization, in: Proceedings of the Third IASTED International Conference on Communications and Computer Networks, Marina del Rey, CA, USA, 2005.

[39] Endace, Dag network monitoring cards, http://www.endace.com/our-products/dag-network-monitoring-cards/ (accessed 2009-07-03).

[40] Napatech, Napatech protocol and traffic analysis network adapter, http://www.napatech.com (accessed 2009-07-03).

[41] Invea-Tech, Combo accelerated nic cards, http://www.invea-tech.com/solutions/packet-capture (accessed 2009-07-03).

[42] Endace, Ninjaprobe 40G1, http://www.endace.com/ninjaprobe-40g1.html (accessed 2009-10-27).

[43] J. Zhang, A. Moore, Traffic Trace Artifacts due to Monitoring Via Port Mirroring, in: E2EMON'07: Workshop on End-to-End Monitoring Techniques and Services, 2007.

[44] V. Paxson, Growth trends in wide-area tcp connections, Network, IEEE 8 (4) (Jul/Aug 1994) 8–17.

[45] P. Phaal, S. Panchen, N. McKee, InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks, RFC 3176 (Informational) (2001).

[46] B.-Y. Choi, J. Park, Z.-L. Zhang, Adaptive packet sampling for accurate and scalable flow measurement, Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE 3 (29 Nov.-3 Dec. 2004) 1448–1452 Vol.3.

[47] T.Zseby, M. Molina, N.Duffield, S.Niccolini, F.Raspall, Sampling and Filtering Techniques for IP Packet Selection, IETF Internet Draft, http://www.ietf.org/internet-drafts/draft-ietf-psamp-sample-tech-10.txt (accessed 2009-07-03).

[48] B. Claise, IPFIX Protocol Specification, IETF Internet Draft, http://tools.ietf.org/html/draft-ietf-ipfix-protocol-21 (accessed 2009-07-03).

[49] C. Estan, G. Varghese, New directions in traffic measurement and accounting, in: SIGCOMM '02: Proceedings of the 2002 conference on Applications, technologies, architectures, and protocols for computer communications, 2002, pp. 323–336.

[50] N. Duffield, C. Lund, M. Thorup, Properties and prediction of flow statistics from sampled packet streams, in: IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment, 2002, pp. 159–171.

[51] E. Cohen, N. Duffield, H. Kaplan, C. Lund, M. Thorup, Algorithms and estimators for accurate summarization of internet traffic, in: IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, 2007, pp. 265–278.

[52] M. C. Caballer, L. Zhan, Compression of internet header traces, Tech. rep., Master Thesis, Chalmers University of Technology, Department of Computer Science and Engineering (2006).

[53] V. Paxson, Strategies for sound internet measurement, in: IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, 2004, pp. 263–271.

[54] J. Cleary, S. Donnelly, I. Graham, A. McGregor, M. Pearson., Design principles for accurate passive measurement, in: PAM '00: Proceedings of the Passive and Active Measurement Workshop, 2000.

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

[56] D. Mills, Network Time Protocol (Version 3) Specification, Implementation and Analysis, RFC 1305 (Draft Standard) (1992).

[57] V. Paxson, On calibrating measurements of packet transit times, in: SIGMETRICS '98/PERFORMANCE '98: Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems, 1998, pp. 11–21.

[58] J. Micheel, S. Donnelly, I. Graham, Precision timestamping of network packets, in: IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, 2001, pp. 273–277.

[59] S. Donnelly, Endace dag timestamping whitepaper, endace, http://www.endace.com/ (2007, accessed 2009-07-03).

[60] A. Pásztor, D. Veitch, Pc based precision timing without gps, in: SIGMETRICS '02: Proceedings of the 2002 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, 2002, pp. 1–10.

[61] E. Technologies, CDMA Network Time Server, http://www.endruntechnologies.com/pdf/TempusLxCDMA.pdf (accessed 2009-07-03).

[62] P. O. Hedekvist, R. Emardson, S.-C. Ebenhag, K. Jaldehag, Utilizing an active fiber optic communication network for accurate time distribution, Transparent Optical Networks, 2007. ICTON '07. 9th International Conference on 1 (1-5 July 2007) 50–53.

[63] S. Moon, P. Skelly, D. Towsley, Estimation and removal of clock skew from network delay measurements, INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (1999) 227–234.

[64] L. Zhang, Z. Liu, C. Honghui Xia, Clock synchronization algorithms for network measurements, INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE (2002) 160–169.

[65] J. Wang, M. Zhou, H. Zhou, Clock synchronization for internet measurements: a clustering algorithm, Comput. Networks 45 (6) (2004) 731–741.

[66] Y. Lin, G. Kuo, H. Wang, S. Cheng, S. Zou, A fuzzy-based algorithm to remove clock skew and reset from one-way delay measurement [internet end-to-end performance measurement], Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE 3 (2004) 1425–1430 Vol.3.

[67] D. DuBois, H. Prade, Fuzzy sets and systems: theory and applications, Academic Press, 1980.

[68] H. Khlifi, J.-C. Grégoire, Low-complexity offline and online clock skew estimation and removal, Comput. Networks 50 (11) (2006) 1872–1884.

[69] C. Shannon, D. Moore, K. Keys, M. Fomenkov, B. Huffaker, k claffy, The internet measurement data catalog, SIGCOMM Comput. Commun. Rev. 35 (5) (2005) 97–100.

[70] M. Allman, E. Blanron, W. Eddy, A scalable system for sharing internet measurement, in: PAM '02: Passive & Active Measurement Workshop, 2002.

[71] CAIDA, DatCat: Internet Measurement Data Catalog, `http://imdc.datcat.org/`.

[72] J. Mogul, Trace anonymization misses the point, WWW 2002 Panel on Web Measurements, `http://www2002.org/presentations/mogul-n.pdf` (2002, accessed 2009-07-03).

[73] J. Mirkovic, Privacy-safe network trace sharing via secure queries, in: Proceedings of the 1st ACM workshop on Network data anonymization, 2008.

[74] A. Parate, G. Miklau, A framework for safely publishing communication traces, in: CIKM '09: Conference on Information and Knowledge Management, 2009.

[75] S. Kang, N. Aycirieix, H. Kwak, S. Kim, S. Moon, CASFI Data Sharing Platform, in: PAM '09: Passive and Active Network Measurement Conference, Student Workshop, 2009.

[76] J. Apisdorf, K. Claffy, K. Thompson, R. Wilder, Oc3mon: Flexible, affordable, high performance staistics collection, in: LISA '96: Proceedings of the 10th USENIX conference on System administration, Berkeley, CA, USA, 1996.

[77] J. Stone, C. Partridge, When the crc and tcp checksum disagree, in: SIGCOMM'00: Conference on Applications, Technologies, Architectures and Protocols for Computer Communication, 2000.

[78] A. Rijsinghani, Computation of the internet checksum via incremental update, RFC 1624 (1994).

[79] W. John, On measurement and analysis of internet backbone traffic, Tech. rep., Licentiate Thesis, Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, SE, ISSN 1652-076X, Technical Report 50L (2008).

[80] W. John, S. Tafvelin, Analysis of internet backbone traffic and header anomalies observed, in: IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, 2007, pp. 111–116.

[81] M. Zhang, M. Dusi, W. John, C. Chen, Analysis of UDP Traffic Usage on Internet Backbone Links, in: SAINT'09: 9th Annual International Symposium on Applications and the Internet, 2009.

[82] W. John, M. Dusi, kc claffy, Estimating Routing Symmetry on Single Links by Passive Flow Measurements, in: under review, 2009, `http://www.caida.org/research/traffic-analysis/asymmetry/`, (accessed 2009-10-27).

[83] M. Zhang, W. John, kc claffy, N. Brownlee, State of the Art in Traffic Classification: A Research Overview, in: PAM '09: Passive and Active Network Measurement Conference, Student Workshop, 2009.

[84] E. Hjelmvik, W. John, Statistical Protocol IDentification with SPID: Preliminary Results, in: SNCNW'09: 6th Swedish National Computer Networking Workshop, 2009.