CHALMERS

Technical Report No. 2008-17

Experiences from passive Internet Traffic Measurements

WOLFGANG JOHN
SVEN TAFVELIN

Department of Computer Science and Engineering
Division of Networks and Systems
CHALMERS UNIVERSITY OF TECHNOLOGY/
GÖTEBORG UNIVERSITY
Göteborg, Sweden, 2008

Experiences from passive Internet Traffic Measurements

Wolfgang John and Sven Tafvelin
Department of Computer Science and Engineering
Chalmers University of Technology, Göteborg, Sweden
{wolfgang.john, sven.tafvelin}@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 challenges discussed will eventually appear when planning to conduct passive measurements on high-speed network connections, such as Internet backbone links. After giving a general overview of network measurement approaches, a summary of different design options and important considerations for backbone measurements is given based on the lessons learned from a successful Internet measurement project. 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 challenges range from handling the vast amounts of network data to timing and synchronization issues. Finally, policies regarding public availability of network data need to be established once data is successfully collected. This paper provides tutorial guidelines for setting up and performing future passive Internet measurement projects.

1 Introduction

Today, the Internet has emerged as the key component for commercial and personal communication. One contributing factor to the ongoing expansion of the Internet is its versatility and flexibility. In fact, almost any electronic device can be connected to the Internet now, ranging from traditional desktop computers, servers or supercomputers to all kinds of wireless devices, embedded systems, sensors and even home equipment. Accordingly, the usage of the Internet changed dramatically since its initial operation in the early 80's, when it was a research project connecting a handful of computers, facilitating a small set of remote operations. Nowadays (2008), 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 media between hundreds of millions of nodes.

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 via IP. 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 protocols and network infrastructure are usually well understood when

tested in isolated lab environments or in network simulations. However, their behavior when observed while interacting with the vast diversity of applications and technologies in the real, hostile Internet environment is often unclear, especially on a 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 within 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 employment 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. However, it is 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 (MonNet, Section 8) 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 addresses the major challenges which will eventually appear when planning to conduct measurements on high-speed network connections. Thus, this paper can be regarded as basic guideline for passive Internet measurement projects. 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), which include access privileges to the network operator's premises and permission to perform installation and maintenance of measurement equipment. Once these legal and operational obstacles are overcome, a third challenge is given by various technical difficulties when actually measuring high-speed links (Section 6). Technical challenges range from handling the vast amounts of network data to timing and synchronization issues. Finally, different considerations regarding public availability of network data are discussed, which should eventually be taken into account once data is successfully collected (Section 7).

1.1 Paper outline

Section 2 gives an overview of different network traffic measurement approaches and methodologies. In Sections 3 - 7 the main challenges encountered while conducting Internet measurements are addressed and discussed. These can be seen as tutorial guidelines for future measurement projects. Section 8 will then briefly outline the MonNet project, which is the measurement project providing the experience for the present paper. Finally, Section 9 will discuss future challenges of Internet measurement and conclude the paper.

2 Internet measurement methodologies

This section gives an overview of general network measurement approches. The basic approaches are categorized among different axes, with the most suitable methods for passive Internet measurements according to current best practice 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). 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 though active measurement offers some possibilities that passive approaches can not provide, in this paper only passive measurement is considered, since it 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 is used to collect 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. Currently, the most common capture cards for high-speed network measurements are Endace DAG cards [6], but also other companies offer such equipment, such as Napatech [7] or Invea-Tech [8].

Once network data is collected, it needs to be processed to fulfill its particular purpose, such as analysis of certain properties. 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 [9]. While link-layer protocols dictate the technology used for the data collection (e.g. SONET/HDLC, Ethernet), the most studied protocol is naturally the Internet Protocol (IP), located on the network layer. The Internet measurement community commonly also shows great interest in 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 [10]. 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 [11], or bidirectional, as used in different studies by MonNet [12, 13, 14]. 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 a maximum of analysis possibilities, they come along with a number of technical and legal issues, as discussed in Chapters 3 - 6. Therefore, it is common practice to reduce the stored information to packet headers up to a certain protocol level, e.g. including network and transport protocol only, as done for the MonNet traces. Such **packet header traces** are an efficient way to reduce processing and storage costs, while at the same time addressing legal and privacy concerns. These advantages come, however, with the drawback of reduced analysis possibilities for Internet applications.

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 (LAN), is the **PCAP format**, the format of the BSD Packet Filter and TCPdump. For traces of wide area networks (WAN), 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 [15] format CRL or NLANR's formats FR, FR+ and TSH. This diverseness in trace formats introduces some problems, since public available analysis tools usually do not recognize all of these formats, making conversion of traces from one format to another necessary. Even tools for direct conversion often do not exist, so it might be necessary to convert traces into PCAP format first, which can be seen as the de-facto standard. Thus almost all conversion tools are able to convert their own format to or from PCAP format. Conversion however is usually not without costs. Different timestamp conventions within the trace formats often lead to loss of timestamp precision, which should be considered when performing timing sensitive operations, such as merging of trace files, or calculation of packet delays or inter-arrival times.

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. While regulations are still strong for protection of user privacy, recent terrorist attacks lead to amendments to both European and US directives and laws. Data retention and network forensics are increasingly gaining legal importance at the expense of user privacy, which is likely to result in further changes of laws in the near future.

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* [16] of 1995 and the complementing *Directive 2002/58/EC on Privacy and Electronic Communications* [17] 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* [18].

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 is 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 is 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 is 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 is 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. On the other hand, 26 countries declared to postpone application of this directive regarding Internet access, Internet telephony and Internet email, which is possible until 14 March 2009 according to Article 15(3).

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

This overview of US privacy laws will follow a recent article by Sicker et al. [19], thereby focusing on federal laws of the US only (as opposed to state laws), especially since they are probably best comparable to the overarching EU directives. There are two relevant sets of federal US laws applying to Internet measurement: one for real-time monitoring, and another 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 [20]), 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 to the communication as well as the providers right to protect his network 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 [21]) 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 [22, 23, 24]). 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 (such as header traces) can be shared with anyone, with exception of the government. This in turn 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, especially 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 [19], 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. [19], often there is a *"disconnect between the law and current academic practice"*. Since laws are not likely to be changed in favor of scientific Internet measurement anytime soon, 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 (e.g. a RFC), as suggested by Sicker et al [19]. 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 6.1). 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.

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) to *keep* raw measurement data *secret*
2) to *de-sensitize* the data, which can be done by one or both of the following ways:
   2a) to *remove packet payload data* in packet-level traces
   2b) to *anonymize* 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 is required would be network forensics conducted by governmental authorities. In this case data is 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 kept internal only, inside a network operator's organization.

The above stated de-sensitization actions, payload removal and anonymization, seem to be good policies which satisfy both data providers (operators) and researchers or 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 TCP/IP headers only, stripping off data after transport headers. While this is a good way to make sure that sensitive payload information is removed, it limits analysis possibilities, especially when application level protocols are to be investigated. 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 headers), since a lot of sensitive information can be found there. Other application headers, such as HTTP or 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 indefinite 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.[25]) 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 of how much sensitive information is included into unknown or undefined application payloads. This remaining uncertainty might be tolerable if traces are only accessed by a limited number of trusted researches, but is unsuitable for traces indented to become publicly available.

Even if the boundary between packet header and packet payload is clearly defined (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, such as CLNS routing updates (Connectionless Network Protocol), CDP messages (Cisco Discovery Protocol) or unknown or malformed transport headers. Such packets could be a) truncated by default after a specified number of bytes; b) dropped entirely, which should be at least recorded in meta-data describing the specific trace; c) kept un-truncated, which might bear security and privacy risks. 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.

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 [26]. 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 removing 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 [27]. In some situations, it might be sufficient to anonymize IP addresses only. 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. [26] also showed that hardware addresses in link layer headers can reveal confident 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 [26, 28, 29, 30]. This was stated nicely by Allman and Paxson in [31], when saying that publisher of network traces "*are releasing more information than they think*"!

Currently, the most common practice is to anonymize IP address information only, which is often sufficient for internal use (i.e. only results, but not the 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 field with individual methods. Such a framework is publicly available and described by Pang et al. in [27]. However, how different fields are modified has to be decided by the researcher or agreed upon in anonymization policies. In the following paragraphs, we will discuss some common methods of how to anonymize the most sensitive information in packet headers, namely IP addresses.

4.2.1 Anonymization methods

IP address anonymization can be defined as the irreversible mapping between the real and the anonymized IP addresses. 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). 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 into the real addresses is lost. This general idea 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. 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. The first popular prefix-preserving anonymization technique was used in *TCPdpriv*, developed by Minshall in 1996 [32]. The prefix preserving anonymization function of TCPdpriv (the '-A50' option) 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.

This drawback was fixed by a Cryptography-based Prefix-preserving Anonymization method, *Crypto-PAn*), described by Xu et al. in 2002 [28]. 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 (e.g. Rijndael). 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 and improved in order to suit specific requirements [33, 34].

4.2.2 Quality of anonymization

Recently, different successful attacks on IP address anonymized traces have been presented [26, 29, 30, 35]. Therefore Pang et. al [27] 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 by the external mapping. Note, however, that this scheme is not preserving 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. [26].

Since Crypto-PAn is widely used today and sets an de-facto standard for trace anonymization, 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 therefore common practice 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 regards payload removal and anonymization, it is still an open question on when these operations should be performed. If a policy or an agreement with the network operator states that network data is 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

As discussed above, network data can contain a number of sensitive and confidential data. Even if datasets are planned to be made public, sensitive information needs to be removed first, which might require intermediate steps involving storage of unprocessed raw data. Thus it is crucial to prevent unauthorized access to 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 could 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. Typically, secure access (e.g. SSH) for a limited number of specified hosts inside an organization's LAN should be enough to remotely maintain and operate measurement hosts. Finally, especially in case of discontinuous measurement campaigns, measurement times should be kept secret to minimize the risk of de-anonymization attacks involving hostile 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 personal is present when access to certain premises is necessary. This dependency on authorized personal makes planning and coordination difficult and reduces flexibility and time-efficiency. Flexibility constraints are further exaggerated by the geographic location of some premises, which 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 highly undesired by network operators and further restricts planning flexibilities of Internet measurement projects.

The above indicated operational difficulties suggest the need of careful planning of measurement activities. Planning should include suitable risk management, such as slack time and hardware redundancy where ever 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 apply 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 hard-reboot machines and access to the system console, independent from operating system status.

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 measurements 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, it is above all a technical challenge. Sometimes, clever engineering is required to overcome different technical difficulties. In the following subsections we will therefore provide discussions about important technical aspects regarding Internet measurement, including strategies to cope with the tremendous data amounts and some considerations of 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 time sensitive correlation of different traffic traces is required, such as during passive delay measurements or when merging network traces measured on links of opposing direction.

6.1 Data amount

The amount of data carried on modern Internet backbone links is not 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 2010. This development will emphasize some bottlenecks in measurement nodes which emerge during a measurement process, such as I/O bus bandwidth, memory capacity or disk storage speed. If high-capacity backbone links operate in full line rate, contemporary I/O bus capacities (e.g 8 Gbit/s theoretical throughput for PCI-X or 16 Gbit/s for PCIe) are hardly sufficient to store 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. And 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. Available SCSI disks provide nominal throughput rates of around 5 Gbit/s (e.g. Ultra-640 SCSI), 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 the pace of higher link rates. 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. Only the sheer data amounts of several GByte/s are not easy to handle and fill up memory buffers quickly. All these considerations did still not take the required storage capacity into account. Longitudinal measurement campaigns, recording several Gigabytes of network data per second, are non-trivial tasks and will eventually be limited by storage capacities.

The above provided discussion clearly highlights that recording of complete packet level traces is not scalable, and strictly limited by hardware performance. 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 packets would be discarded, resulting in incomplete, biased datasets, which is highly undesirable with respect to the accuracy of scientific results. As a result, measurement of complete packet level traces is technically not always feasible. In the following paragraphs some approaches aiming to reduce data amounts by still preserving relevant information are presented. Afterwards, some considerations of how to archive large network datasets are provided.

6.1.1 Traffic data reduction techniques

If network data is 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 [36]. 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, 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 [6]).

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 [37]. Alternatively, also more sophisticated packet sampling approaches have been proposed, such as adaptive packet sampling [38]. A good overview of sampling and filtering techniques for IP packet selections can be found in a recent Internet draft by Zseby et al. [39].

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 network infrastructure itself, which explains why the most common flow record format IPFIX (derived from NetFlow) [40] was originally developed by Cisco. 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 [41] and other advanced sampling techniques, such as proposed in [38, 42, 43].

Finally, 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 are truncated after a specified packet length of N bytes, and removal of payload parts is done during offline processing of the traces.

6.1.2 Archiving of network data

Since measuring Internet traffic is a laborious and expensive task, measurement projects typically want to archive not only their analysis results, but also the raw data, such as packet level traces or flow data. Archiving raw data is furthermore important to keep scientific results reproducible, to allow comparisons between historical and current data, to make additional analysis regarding different aspects possible, and finally to share datasets with the research community, as discussed in Section 7.

Archiving of network traces is not always a trivial task, especially for longitudinal, continuous measurement activities. Description of different archiving solutions is not within the scope of this paper, but it should be mentioned that such solutions, automatic, semi-automatic or manual, need to be carefully engineered, including risk management such as error handling and redundancy. Data redundancy can be provided by suitable RAID solutions or even 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 [44]. When network data is archived, it is also crucial to attach descriptive meta-data to datasets, as argued by Pang et al., Paxson, and Cleary et al. [27, 45, 46]. 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. [27] recommend to associate meta-data to datasets by adding a checksum digest of the trace to the meta-data file.

6.2 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 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 of relevant measurement hard- and software or ensuring timestamps consistency. Furthermore, an early basic analysis of traces can reveal unanticipated errors, which might require manual inspection. Statistical properties or 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 exemplary sanitization procedure is described in John, Section 5.3.2 [47]. Another example of an automated sanitization process is provided by Fraleigh et al. in [48], and a more general discussion about sanitization can be found in Paxson's guidelines for Internet measurement [45].

6.3 Timing issues

Internet measurement has an increasing need for precise and accurate timing, considering that 64 byte sized packets (e.g. minimum length Ethernet frame) traveling back to back on 10 Gbit/s links arrive with as little as 51 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 or delay measurements. Before discussing different timing and synchronization issues involved into 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 used in Internet measurement, this subsection is finished by giving a discussion of how accurate timing and clock synchronization can be provided.

6.3.1 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). UTC is derived from the average of more than 250 Cesium-clocks situated around the world. A perfect clock would report true time, according to UTC at any given moment, thereby providing a constant rate. The clock terminology definitions provided below follow Mills' network time protocol (NTP) version 3 standard [49] and the definitions given by Paxson in [50].

- 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.

6.3.2 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 packets. 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 packets arriving back to back. These error sources are typically not an issue for hardware solutions, such as Endace DAG cards [6]. 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 [51].

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 [49], 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 therefore can store timestamps with a resolution of 232 pico seconds. Currently, in the most advanced hardware can actually use 27 bits of the fraction part, providing a resolution of 7.5 ns [52]. Future improvements of clock resolutions will therefore cause no modifications in timestamp or trace formats. 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 time precision from nanosecond to microsecond scale.

6.3.3 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 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 [53] 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 a nanosecond resolution, but above all highly improved clock stability, with a skew similar to GPS synchronized solutions.

Since tight synchronization is of increasing importance, modern network measurement hardware incorporates a special timing systems, such as the DAG universal clock kit (DUCK) [51, 52] 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 back to back 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 2 clock ticks. For 134 Mhz oscillators this consequently means an accuracy of $\pm$ 15ns, which can be regarded as very high clock stability.

6.3.4 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 accuracy 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 to 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 often used to adjust one-way-delay measurements involving remote measurement locations.

6.3.4.1 Continuous clock synchronization

There are different ways how clocks can be synchronized. The most common way to synchronize a clock of a computer to a time reference is the network time protocol NTP [49]. 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 [50]. These observations lead to the recommendation to disable NTP synchronization during measurement campaigns, thus providing NTP synchronization only during the time before and after measurement intervals.

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 on earth can then pick up the signals from multiple satellites, which allows calculating the current position of the receiver relative to the satellites by estimating the distances and triangulation. 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. This means that GPS antennas should 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.

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 [54]. Base stations of cellular networks are all equipped with GPS receivers to retrieve timing information. This information is then broadcasted as control signal within the network. Since base stations operate on lower frequencies, it is possible to use these base stations as timing source 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 base station for various different reasons.

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 [55]. 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.

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 [52]. 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.

6.3.4.2 Retrospective time correction

In some cases, where measurements timestamped by different clocks need to be compared, accurate clock synchronization cannot be provided. It might also be the case that synchronization accuracy is simply not sufficient (e.g. when using NTP). Therefore, a number of algorithms to compensate for errors have been

proposed. These algorithms are especially useful to correct estimations of transit times or end to end delays, which often involves measurement locations with large geographical distances. Various interesting methods for retrospectively removing of offset and skew from delay measurements have been proposed during last ten years, such as [50, 56, 57, 58, 59, 60].

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 or research groups have the possibilities to collect Internet data, which makes proper traces a rare resource. Therefore, the Internet measurement community has repeatedly been encouraged to share their valuable datasets and make them publicly available [61, 45, 31], given that sharing of network data is legally permitted (see Section 3). Sharing network data is not only a service to the community, it is also an important factor when it comes to credibility of research results. Sharing and archiving of data is therefore fundamental to scientific progress and helps to improve scope and quality of future research, as highlighted in other research fields as well [62].

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 previous 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 [31], Allman and Paxson discuss the above mentioned benefits of data availability, including ethic and privacy considerations, as discussed in Section 4.

An alternative approach to data sharing was suggested by Mogul in a presentation in 2002 [63]. He proposes a 'move the 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.

In any case, a prerequisite for either of the above mentioned approaches is that researchers are made aware of existing and available datasets. A system for sharing Internet measurements was proposed by Allmann in 2002 [64]. This system was inspiration for CAIDA to finally implement the Internet measurement data catalog DatCat [65], which allows publication of meta-data about network datasets. The goal of this project is to provide the research community with a central database, providing searchable descriptions of existing datasets. DatCat was opened to public viewing during 2006, and currently allows contributions of trace descriptions by invitation only. The vision of this pioneering project is to eventually allow contributions of anyone and thereby providing a recognized and commonly used platform for sharing of measurement data.

8 Example: The MonNet project

This chapter provides the procedure description of one project for passive Internet traffic monitoring and analysis: the MonNet project. This project is also the source for many of the experiences presented in this guideline paper.

Before network measurements could be started, a number of preparatory steps needed to be performed. First, MonNet, as a project regarding Internet and traffic measurements and analysis, was proposed to the SUNET (Swedish University Network) board. In order for the project to be granted, the SUNET board required permission of 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, where one of these boards is responsible for the region of Gothenburg. After two meetings and elaborate discussions about the de-sensitization process of the traces, the regional ethics committee finally permitted the MonNet measurements to take place. The provided permission from the research ethics committee can be regarded as an appropriate safeguard for measurement of Internet traffic for scientifc purposes, as requested by current EU directives (see Section 3.1).

As a next step a measurement location was chosen on the SUNET Internet backbone links between the region of Gothenburg and the central Internet exchange point in Stockholm. The choice was in the first place made to be able to obtain traces of data transferred between a regional network and the main Internet. The chosen location has 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, since the remote management cards the two measurement nodes have been equipped with, turned out to be unstable and unreliable. As a result, a number of physical visits at the measurement location have been necessary due to some unexpected hardware defects. However, 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.

Finally, the measurement and processing nodes applied have been planned and 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 OC3MON/OC48MON [66] and Sprint's IPMON [48], provided valuable inspiration. It should be pointed out that besides the purchasing of basic server equipment, the selection of measurement cards (Endace DAG cards) and suitable optical splitters, both by no means commodity hardware, deserves special attention during the planning phase and usually also requires longer product delivery times. Details about the final measurement equipment can be found in Part II of [47]. The measurement procedure together with meta data of the resulting measurements is documented on CAIDA's DatCat [67].

Even if the preparatory tasks could be listed here very briefly, it is worth mentioning that they turned out to be very time consuming. The MonNet project was proposed to the SUNET board in summer 2004. After a waiting period for legal permission by the ethics committee, problems with delayed delivery of crucial equipment and unexpected early hardware failures, the measurement nodes were not in place and operational before fall 2005, more than one year after the project kick-off. Thereafter it took another six months to gain experience and know-how in conducting sound Internet measurement, when in April 2006 finally the first usable dataset could be collected. Since then, a number of studies have been presented [12, 13, 14, 68, 69], revealing current characteristics of Internet traffic, highlighting anomalous and inconsistent traffic behavior and presenting valuable traffic decomposition beyond transport layer.

9 Future Challenges and Conclusions

The development of the Internet has without doubt not come to an end yet. 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 more and more a 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, to keep up with link speeds will not be the only challenge for Interent measurements. There are a number of both technical and commercial reasons which direclty benefit from results of Internet measurements 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. This means that the Internet community will have an increasing need in 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 form 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 [18], 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 soon need to establish the possibilities to execute this directive. Some states already ratified even more stringent laws, such as the infamous '*FRA law*' in Sweden [70]. The FRA law allows the 'Swedish national defense radio establishment' (*Swedish Försvarets radioanstalt, FRA*) to intercept all Internet exchange points that exchange traffic crossing Swedish borders. 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. An important lesson learned from the MonNet project is that many unforeseen problems will occur during a measurement project. Many of the problems can be avoided by careful and anticipatory planning, including basic risk management such as providing for slack times throughout the project duration. To facilitate the setting-up of future measurement projects, this paper is intended to serve as a guide through the practical issues of Internet measurements. The paper addressed 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 tutorial paper should provide researchers and practitioners with useful guidelines to setting up future monitoring infrastructure - which will in turn help to improve results of traffic analysis and therefore contribute to a better understanding of how the Internet functions in detail.

Acknowledgements

This work was supported by SUNET, the Swedish University Computer Network. The authors furthermore want to thank Tomas Olovsson for valuable discussions and comments throughout the MonNet project.

References

[1] R. Nelson, D. Lawson, P. Lorier, Analysis of long duration traces, SIGCOMM Comput. Commun. Rev. 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 080909).

[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] Endace, Dag network monitoring cards, `http://www.endace.com/our-products/dag-network-monitoring-cards/` (2008).

[7] Napatech, Napatech protocol and traffic analysis network adapter, http://www.napatech.com (2008).

[8] Invea-Tech, Combo accelerated nic cards, http://www.invea-tech.com/solutions/packet-capture (2008).

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

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

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

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

[13] 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.

[14] 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.

[15] 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.

[16] 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).

[17] 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).

[18] 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).

[19] 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.

[20] 18 united states code §2511, http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00002511----000-.html.

[21] 18 united states code §3127, http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00003127----000-.html.

[22] 18 united states code §2701, http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00002701----000-.html.

[23] 18 united states code §2702, http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00002702----000-.html.

[24] 18 united states code §2703, http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00002703----000-.html.

[25] 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.

[26] 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.

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

[28] 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, DC, USA, 2002, pp. 280–289.

[29] 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`.

[30] T. Kohno, A. Broido, K. C. Claffy, Remote physical device fingerprinting, IEEE Trans. Dependable Secur. Comput. 2 (2) (2005) 93–108.

[31] 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.

[32] G. Minshall, Tcpdpriv: Program for eliminating confidential information from traces, `http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html`.

[33] 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.

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

[35] 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.

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

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

[38] 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.

[39] 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`.

[40] B.Claise, IPFIX Protocol Specification, IETF Internet Draft, `http://tools.ietf.org/html/draft-ietf-ipfix-protocol-21`.

[41] 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.

[42] 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.

[43] 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.

[44] 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).

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

[46] 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.

[47] 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).

[48] 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.

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

[50] 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.

[51] 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.

[52] S. Donnelly, Endace dag timestamping whitepaper, endace, `http://www.endace.com/` (2007).

[53] 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.

[54] E. Technologies, CDMA Network Time Server, datasheet, `http://www.endruntechnologies.com/pdf/TempusLxCDMA.pdf` (2007).

[55] 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.

[56] 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.

[57] 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.

[58] 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.

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

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

[61] 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.

[62] R. Rockwell, R. Abeles, Guest editorial: Sharing and archiving data is fundamental to scientific progress, The Journals of Gerontology: Series B Psychological sciences and social sciences 53B (1998) 5–8.

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

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

[65] CAIDA, DatCat: Internet Measurement Data Catalog, `http://imdc.datcat.org/` (accessed 080909).

[66] 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.

[67] W. John, S. Tafvelin, SUNET OC 192 Traces (collection), `http://imdc.datcat.org/collection/1-04L9-9=SUNET+OC+192+Traces` (accessed 080909).

[68] 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.

[69] W. John, T. Olovsson, Detection of malicious traffic on backbone links via packet header analysis, Campus Wide Information Systems 25 (2008) in Press.

[70] Swedish Ministry of Defence, Governmental proposition 2006/07:63: En anpassad försvarsunderrättelseverksamhet, `http://www.regeringen.se/sb/d/108/a/78367` (2007).