THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

On Measurement and Analysis of Internet Backbone Traffic

WOLFGANG JOHN

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

On Measurement and Analysis of Internet Backbone Traffic
Wolfgang John

Technical Report 50L
ISSN 1652-876X
Department of Computer Science and Engineering
Division of Networks and Systems

Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 GÖTEBORG, Sweden
Phone: +46 (0)31-772 10 00

Contact Information:
Wolfgang John
Division of Networks and Systems
Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 GÖTEBORG, Sweden

Phone: +46 (0)31-772 16 74
Email: wolfgang.john@chalmers.se
URL: http://www.chalmers.se/cse/EN/people/john-wolfgang

Printed by Chalmers Reproservice
Göteborg, Sweden, 2008

On Measurement and Analysis of Internet Backbone Traffic

Wolfgang John

Division of Networks and Systems, Chalmers University of Technology

Thesis for the degree of Licentiate of Engineering, a Swedish degree between M.Sc. and Ph.D.

ABSTRACT

In the last decade, the Internet emerged undoubtedly as the key component for commercial and personal communication. The success of the Internet is mainly based on its versatility and flexibility, allowing for the development of network applications ranging from simple text based utilities to complex systems for e-commerce and multi-media content. The ongoing expansion of the Internet is the cause of continuous unitlization and traffic behavior changes. Due to this diversity and the fast changing properties *the Internet is a moving target*. At present, the Internet is far from being well understood in its entirety. However, constantly changing Internet characteristics associated with both time and location make it imperative for the Internet community to understand the nature and behavior of current Internet traffic in order to support research and further development.

Through the measurement and analysis of traffic the Internet can be better understood. This thesis presents a successful Internet measurement project, providing guidelines for conducting passive network measurements. Recent large-scale backbone traffic data is analyzed, revealing current deployment of protocol features on packet and flow level, including statistics about anomalies and misbehavior. A method to classify packet header data on transport level according to network application is proposed, resulting in a complete traffic decomposition. A comparison of the signaling behavior of the main traffic classes - Web, P2P, and malicious traffic - is presented. The results are significant because of the over-all impact of these traffic classes on Internet traffic behavior. The scale of the measurements allows to highlight longitudinal trends and changes in network application and protocol usage. Such findings support pro-active measures such as refinement of network design, provisioning, accounting and security measures. Finally, the analysis of data taken on vital Internet backbone links also provides valuable input for simulation models. By presenting a snapshot of current traffic composition and characteristics, this thesis contributes to a better understanding of how the Internet functions.

Keywords: Measurement, Passive, Internet Backbone, Packet Header, Traffic Analysis, Classification, Connection Behavior, Peer-to-Peer, Malicious Traffic, Header Anomaly

Acknowledgments

First and foremost, I want to express my deepest gratitude to my supervisor and examiner Prof. Sven Tafvelin. Not only did he offer this challenging and interesting position to me, but he has also been a kind and patient adviser, keeping me on track with my education and research. I am also greatly indebted to Assistant Prof. Tomas Olovsson, my co-supervisor, for many encouraging discussions and valuable reviews of my scientific papers. For putting my research progress into perspective I want to thank my 'mentor' Ana Bove, who managed to lift my spirits after our monthly lunch meetings.

For a measurement project like MonNet, which is to a high degree dependent on suitable hardware, it is of crucial importance to keep the systems well configured and running. For this reason, a big thanks goes out to the technical support group at the Department of Computer Engineering for their generous assistance. I am especially grateful to Mr. Pierre Kleberger, who really backed me up when it came to all kinds of technical issues during the first year of my career at Chalmers. I am also very thankful for the continuous support of the kind and helpful administrative staff at the department, who made most administrative obstacles 'magically' disappear, thereby helping to keep my focus on my research and education.

I have the great fortune of working in a place not only with co-workers, but with real friends. I want to thank 'Mr. Andersson' for being a invaluable guide through the streets and the night-life of Göteborg and other European cities. I am also very thankful to Magnus A., not only for teaching me how to enjoy a Cafe Latte, but also for being one of my favorite discussion partners for random topics within both research and everyday life. I want to thank 'Tomtenisse' Ulf L. for sacrificing his career as a rock-star just to be a great colleague and true friend to me here at Chalmers. Both being good friends, many thanks also go to the 'dynamic duo' on the fourth flour for helping me out on various issues. Magnus S. is furthermore a good and reliable companion in all kinds of outdoor activities, and Martin T. deserves my distinct respect for his dedication to organizing social events among colleagues.

In this context, I also want to thank all current and former PhD students and guest students at CSE for being helpful and supportive every single day, making my life at Chalmers a real pleasure. These colleagues include Dennis N., Anders N., Djordje J., Vilhelm V.,

Minh Q. D., Mafijul I., Waliullah M. M., Mindaugas D., Marius G., Marco G., Ana B. and Jochen H. Additionally, I really feel fortunate to work in such a pleasant and inspiring environment, and for this I finally want to thank all people at the Department of CSE.

When moving alone to a foreign country as PhD student, it is not always easy to establish a social environment outside the University world. I had the great luck to meet a number of very welcoming and nice people early on, who really made me feel 'at home' in Göteborg in no time. It seems impossible to provide an exhaustive list here, so I simply want to express my deepest gratitude to all the extraordinary people who 'hang out' with me in my spare time, thereby making my stay in Sweden so memorable and enjoyable! *'Tusen Kamelåså!'*

I furthermore feel very privileged to have fantastic friends back home in Austria and all over the world, who always remain in touch and remember me even during long time periods without any contact. It is a real fortune to have such amazing people in my life - people I can always count on!

I also should not forget about the people who inspired and supported me before I really started my career as PhD student. Hereby, I want to thank my friends and collaborators during my Master's study in Halmstad, especially Philipp N. and Andreas F. It was during our fruitful and efficient teamwork when I discovered my passion for research, which finally lead me to my current position at this department. I am also indebted to Florian Otel and Shiva Ramagopal for their preparatory work in the MonNet project and their valuable support in my first weeks at Chalmers, which enabled a smooth and comfortable start to my post-graduate career.

I would like to thank my family in Austria sincerely for their support and understanding during all of my life and for keeping me grounded and connected to my beautiful home, in the center of the Alps. Herzlichen Dank liebe Mum, Dad und die restlichen John-Sohns!

This research is sponsored by SUNET, the Swedish University Computer Network. Furthermore, I want to thank SUNET for reliable assistance with both operational and technical issues. I also want to send my acknowledgments to KBM, the Swedish Emergency Management Agency, for financially supporting a part of this work.

Wolfgang John
Göteborg, February 2008

List of appended papers

The papers presented in this thesis, as listed below, have been accepted to conferences:

Paper I: **Wolfgang John** and Sven Tafvelin, "Analysis of Internet Backbone Traffic and Anomalies observed" in *IMC '07: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, San Diego, California, USA, 2007

Paper II: **Wolfgang John** and Sven Tafvelin, "Differences between in- and outbound Internet Backbone Traffic" at *TNC '07: TERENA Networking Conference*, Copenhagen, DK, 2007.

Paper III: **Wolfgang John** and Sven Tafvelin, "Heuristics to Classify Internet Backbone Traffic based on Connection Patterns" in *ICOIN '08: Proceedings of the 22th International Conference on Information Networking* (Proceedings published by the IEEE Communications Society), Busan, Korea, 2008.

Paper IV: **Wolfgang John**, Sven Tafvelin and Tomas Olovsson, "Trends and Differences in Connection Behavior within Classes of Internet Backbone Traffic" at *PAM '08: the 9th Passive and Active Measurement Conference* (Proceedings to be published in the Springer Lecture Notes in Computer Science), Cleveland, Ohio, USA, 2008.

Contents

Part I

INTRODUCTION

1 Thesis objectives

1.1 Motivations for measuring the Internet

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 (handhelds, mobile phones, etc.), embedded systems, sensors and even home equipment (entertainment consoles, major appliances, etc.). Accordingly, the usage of the Internet changed dramatically since its initial operation in 1969, when it was a research project connecting a handful of terminals, thereby 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.

Unfortunately, this rapid development has left little time or resources to integrate measurement and analysis possibilities into Internet infrastructure, applications and protocols. 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 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 'shape' 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 (consequently short for *INTERconnected computer NETworks*). Thus Internet protocols and applications are not only changing with time, but also within geographical locations. Furthermore, increasing bandwidths and growing numbers of Internet users also lead to increased misuse and anomalous behavior [1], which needs to be studied in order to develop suitable counter strategies. Overall, this means that even though the the Internet may be considered to be the most important modern communication platform, a number of questions concerning how and why it functions remain unanswered. For example, many network operators can not answer the fundamental question of which traffic is carried on their networks due to the disguising properties of newly emerged peer to peer (P2P) file sharing applications. In order to support research and further development of the Internet, it is crucial that the Internet community understands the nature and detailed behavior of contemporary Internet traffic.

The best way to acquire a better and more detailed understanding of the modern Internet is to measure real Internet traffic, preferably on highly aggregated links. Unfortunately, measuring Internet traffic is not simple and involves a number of challenging tasks, as discussed in Chapter 3. However, once the technical, practical and legal complications of Internet measurement are overcome, the results will assist in the improvement of network design and provisioning, robustness of protocols and infrastructure, network performance and accounting. Furthermore, Internet measurements reflecting network behavior as seen 'in the wild' provide not only much-needed input for refinement of simulation models [2], they also support further development of value-added services (Quality of Service (QoS), Voice over IP (VoIP), etc.) and improvement of security and intrusion detection systems. Ongoing measurements will reveal longitudinal trends and changes in the usage of network applications and protocols, thus allowing the network research and development community to remain pro-active.

1.2 Thesis objectives

As described in Section 4, different Internet measurement projects have been carried out in the past and some are still active. However, large scale backbone measurements are rare, therefore a substantial number of relevant research questions are still unanswered. The Internet is far from being well understood in its entirety, especially since Internet characteristics vary according to time and location - *the Internet is a moving target*! The MonNet project, described in Section 5, sets out to provide a better understanding by presenting current characteristics of Internet traffic based on a large amount of empirical data. The resulting datasets thereby contribute to the current global understanding of the Internet. More precisely, this thesis has the following objectives:

- **Providing guidelines for passive Internet measurement**
  - Discussing challenges of Internet measurement, resulting in guidelines to passive traffic measurements based on experience and lessons learned (Chapter 3)
  - Presenting an example of a successful measurement project on an Internet backbone (Chapter 5)
- **Revealing current characteristics of Internet traffic**
  - Revealing deployment of protocol specific features on packet level (Papers I and II)
  - Presenting behavioral differences of flows with respect to direction and application (Papers II and IV)
  - Providing detailed input for refinement of future Internet simulation models (Papers I, II, IV)
- **Presenting traffic decomposition beyond transport layer**
  - Proposing a simple and fast transport-level classification method of backbone traffic, disregarding packet payloads (Paper III)
  - Highlighting longitudinal trends and diurnal differences in connection behavior within classes of Internet applications (Paper IV)
- **Highlighting anomalous and inconsistent traffic behavior**
  - Highlighting incorrect implementations and invalid use of protocols on packet level to support improvement and optimization of protocols (Papers I and II)
  - Studying quantity and behavior of malicious and attack traffic in order to support design and refinement of necessary security measures (Papers I, II, IV)
  - Revealing directional differences in backbone traffic, showing that malicious Internet traffic is not uniformly distributed (Paper II)

1.3 Thesis outline

After the discussion explaining the general motivation for conducting Internet measurement and the specific objectives of this thesis, *Part I* is continued by providing relevant background information about Internet measurement. Chapter 2 gives an overview of different network traffic measurement approaches and methodologies. In Chapter 3, challenges encountered while conducting Internet measurements are addressed and discussed. These can be seen as basic guidelines for future measurement projects. Apart from technical aspects, the challenges include legal and ethical questions as well as issues about public availability of network logs and traffic traces. Finally, Chapter 4 gives an overview of some prominent Internet measurement projects and other relevant work done within the scope of this thesis.

Part II presents the MonNet project. The MonNet project is a project for passive Internet measurement and analysis carried out at the Department of Computer Science and Engineering at Chalmers University. MonNet, forming the basis for this thesis, is described in Chapter 5, which includes descriptions of the technical solution for the measurement nodes performing data collection, the measurement process and different analysis tools developed within the project. Next, the scientific contributions of the MonNet project are pointed out by giving short summaries of the papers included. The final chapter provides an outlook for the future and outlines upcoming opportunities.

Part III provides the main scientific contribution of this thesis, by including four papers which have either been published or accepted for publication at recognized scientific conferences upon completion of this thesis. The above mentioned thesis objectives (Section 1.2) are covered in one or several of the included papers.

2

Internet measurement methodologies

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, 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 theses 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 [3]). While this approach is inexpensive and offers good adaptability, its possibilities to measure traffic on high speed networks are limited. 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. The measurements described in this thesis are performed by the use of optical splitters feeding Endace DAG cards [4], currently the most common capture cards for high-speed network measurements on the market.

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 is 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 the obvious choice for complex and time consuming Internet analysis, as the type of analysis carried out in this thesis.

Internet measurement can furthermore operate on different protocol layers, following the Internet reference model [5]. 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, esp. TCP and UDP. Some Internet measurement projects even have the possibilities to study all layers, including application layer protocols. In this thesis, we will mainly consider network and transport layer protocols.

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 [6]. 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 [7], or bidirectional, as used in Papers II-IV included into this thesis. The most fine 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 Chapter 3. 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 traces described in this thesis. 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. The traces analyzed in this thesis have been recorded in ERF format. Other trace formats seen in the Internet measurement community include CAIDA's CORALReef [8] 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

Challenges when measuring Internet traffic

In this chapter, some major challenges are addressed, which will eventually appear when planning to conduct measurements on high-speed network connections, such as Internet backbone links. Thus, the chapter can be regarded as basic guidelines 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 authorization to traffic collection is granted (Sections 3.1 and 3.2). Second, operational difficulties need to be solved (Section 3.3). These include access privileges to the network operator's premises and permission to perform installation and maintenance of measurement equipment. Such operational issues are especially cumbersome in cases where measurements are carried out on external network infrastructures. Once these legal and operational obstacles are overcome, a third challenge is given by various technical difficulties when actually measuring high-speed links (Section 3.4). 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 3.5).

3.1 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.

3.1.1 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* [9] of 1995 and the *Directive 2002/58/EC on Privacy and Electronic Communications* [10] of 2002, which complements Directive 95/46/EC. 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* [11].

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 this conditions, as long as appropriate safeguards for this data are provided by individual member states.

The e-privacy directive (Directive 2002/58/EC) is complementary to 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, defence, 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 3.4). Experiences and lessons learned from scientific Internet measurement projects are therefore vital and further underline the relevance of Internet measurement.

3.1.2 United States (US) laws

This overview of US privacy laws will follow a recent article by Sicker et al. [12], 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 [13]), 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 [14]) 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 [15–17]). Basically, it is prohibited for network providers to give away stored records of network activity, regardless wether 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.

3.1.3 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 juristic amateurs such as typical network scientists. Common privacy protection measures have been surveyed on datasets used in 57 recent Internet mea-

surement related articles in [12], showing that a majority of network traces was 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 3.2.

As pointed out by Sicker et al. [12], 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 [12]. 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 3.2 and 3.4.1). 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.

3.2 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 measured. 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 even 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 operators 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 included into often imprecise and broadly stated policies. Some important considerations are discussed below.

3.2.1 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 research on application level is intended. One could argue that application headers are technically not user payload, and therefore could be kept as well. This is of course problematic in some case (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. [18]) 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 [19]. 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.

3.2.2 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 [20]. 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. [19] 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 [19, 21–23]. This was stated nicely by Allman and Paxson in [24], 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 [20]. 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.

Anonymization methods

IP address anonymization can be defined as 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 [25]. 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 [21]. 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 anoymization method, and has since been modified and improved in order to suit specific requirements [26, 27].

Quality of anonymization

Recently, different successful attacks on IP address anonymized traces have been presented [19, 22, 23, 28]. Therefore Pang et. al [20] 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 address 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. [19].

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 straightforward to re-translate anonymized addresses bit by bit, which opens for a complete de-anonymization of the trace. The most safe 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. On the other hand, as long as further traces are planned to be anonymized in a consistent manner, destruction or loss of the key would also be unfavorable, which can be solved by a suitable, secure backup solution.

3.2.3 How long to store?

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 allow 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'? 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.

3.2.4 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, SSH access 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.

3.3 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.

3.4 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 considera-

tions of how to get confidence in the measured data. Finally, we will discuss the important challenge of timing and synchronization, which is an important issue in network measurement, especially when time sensitive correlation of different traffic traces is required.

3.4.1 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, 40Gbit/s links already operational and 100Gbit/s links planned to be introduced until 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) are not 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. But even if the I/O bus bottleneck could be overcome, speed of storage array systems would not suffice. Modern storage array network (SAN) solutions offer in the best case 10Gbit/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-0 disk arrays. 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 a non-trivial task and will eventually be limited by storage capacities.

The above provided discussion clearly highlights that measurement 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 uncomplete, biased datasets, which is highly undesirable with respect to the accuracy of scientific results. Obviously, measurement of complete packet level traces is technically not

always feasible. In the following paragraphs we will therefore present some approaches aiming to reduce data amounts by still preserving relevant information. Afterwards, we will provide some considerations of how to archive large network datasets.

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 [29]. 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 [3]), 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 [4]).

Another method to reduce data amounts of packet level traces is packet sampling. Packet sampling can be done systematic, in static intervals (record every Nth packet only) or in random intervals, like proposed by sFlow [30]. Alternatively, also more sophisticated packet sampling approaches have been proposed, such as adaptive packet sampling [31]. A good overview of sampling and filtering techniques for IP packet selections can be found in a recent Internet draft by Zsebz et al. [32].

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) [33] 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 [34] and other advanced sampling techniques, such as proposed in [31, 35, 36].

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 3.2.1, 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 an offline processing of the traces.

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 reproduceable, 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 3.5.

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 thesis, 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 provide 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 [37]. When network data is archived, it is also crucial to attach descriptive meta-data to datasets, as argued by Paxson, Pang et al. and Cleary et al. [20, 38, 39]. 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. In order to avoid confusion, Pang et al. [20] recommend to associate meta-data to datasets by adding a checksum digest of the trace to the meta-data file.

3.4.2 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 this thesis (Section 5.3.2). Another example of an automated sanitization process is provided by Fraleigh et al. in [40], and a more general discussion about sanitization can be found in Paxsons guidelines for Internet measurement [38].

3.4.3 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 10Gbit/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.

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 [41] and the definitions given by Paxson in [42].

- 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.

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 [3], 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 [4]. 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 [43].

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 [41], 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 26 bits of the fraction part, providing a resolution of 15 ns. 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.

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 [44] 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) [43, 45] in Endace DAG cards. The most advanced DUCK clocks currently run at frequencies of 67 Mhz, providing a resolution of 15 ns, which is sufficient for back to back packets on 10Gbit/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 67 Mhz oscillators this consequently means an accuracy of $\pm$ 30ns, which can be regarded as very high clock stability.

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 might 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.

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 [41]. 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.

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 [46]. 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 [47]. 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 [45]. 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, whereas accuracy with respect to true time (UTC) is no major concern.

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 [42, 48–52].

3.5 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 [24, 38, 53], given that sharing of network data is legally permitted (see Section 3.1). 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 and helps to improve scope and quality of future research. Even though from a completely different research field, Rockwell and Abeles provide an interesting and relevant discussion about reasons why sharing and archiving of data is fundamental to scientific progress, which can be found in [54].

Sharing network traces adds reliability to research, since it makes results reproduceable 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 an 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 [24], Allman and Paxson discuss the above mentioned benefits of data availability, including ethic and privacy considerations, as discussed in Section 3.2.

An alternative approach to data sharing was suggested by Mogul in a presentation in 2002 [55]. 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 first proposed by Allmann in 2002 [56]. This system was inspiration for CAIDA to finally implement the Internet measurement data catalog DatCat [57], 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 Internet measurements.

4

Related work on passive Internet measurement

Even though large-scale Internet measurement is still rare, there has been a significant amount of effort expended on different Internet measurement activities in recent years. These activities include development of active and passive measurement methodologies and tools, targeting aspects such as network performance, traffic classification and quantification, reliability and security. A complete survey of these measurement activities is outside the scope of this thesis. However, the following paragraphs will give an overview of measurement projects dealing with passive collection of Internet traces and packet-level analysis thereof, which is in close relation to the topic of this thesis. Generally, access to packet-level backbone traces is very uncommon, and a lot of research is performed on relatively small set of publicly available, but somewhat outdated network traces. This overview first presents the most prominent passive measurement projects, which have the possibilities to overcome the challenges of Internet measurement (see Chapter 3) and therefore have access to own measurement facilities and resulting packet-level traces. Second, some smaller traffic analysis projects are pointed out, which are typically depending on shared, thus outdated datasets or flow-level data from cooperating service providers.

Of the six large measurement project presented, two have been already been terminated (NLANR PMA and SCAMPI/LOBSTER). Of the remaining four, two projects have ac-

cess to large-scale distributed measurement infrastructures, but the focus of their analysis is outside the scope of this thesis (SPRINT focuses on wireless systems, data mining and security; WIDE MAWI is mainly focusing on IPv6 and DNS measurements). The WAND network research group and CAIDA on the other hand are still active and partly target similar research questions as the ones covered by this thesis, and have consequently been very valuable sources of inspiration for identification of relevant research questions. Both research groups have been active in a long time, which means that much of their basic research was carried out some years ago. Since the Internet is subject to very fast changes, some of their work needs to be revised or updated. The MonNet project, as described in this thesis, provides new, contemporary data from a different location of the Internet. The novelty of the data together with the high aggregation level of the measured links and the packet-level granularity of the traces contributes to the global picture of the current Internet. The MonNet project not only sets out to update outdated measurement and analysis results, it also complements previous research activities by studying novel, previously unexplored aspects of Internet characteristics.

WAND network research group

The WAND network research group [58] is located at the University of Waikato Computer Science Department. WAND is a real network measurement research group, performing among other things collection of very long trace sets, network analysis, development of analysis software and network simulation and visualization. In the field of passive network measurements, WAND is best known for the WITS archive and the development of the DAG measurement cards. The Waikato Internet Traffic Storage archive (WITS archive [59]) contains about 200GB of traces taken on different locations starting in 1999. Currently, only statistical summaries of the traces are publicly available, but the traces are planned to be shared in the near future. The DAG measurement cards have been developed at WAND as flexible and efficient hardware solutions for network measurements. Nowadays, support and development of DAG equipment is done by Endace [4], founded in 2001 as spin-off company. Except publications describing the development of DAG cards, WAND also contributed scientific measurement results based on WITS data traces, like the analysis of long duration traces by Nelson et al. [60].

CAIDA

The Cooperative Association for Internet Data Analysis (CAIDA) [61] was launched in 1997 and is based at the University of California's San Diego Super Computer Center. CAIDA sets out to provide tools and analyses in order to promote maintenance of a robust,

scalable global Internet Infrastructure. The broad research activities include routing and addressing, topology, DNS, security, performance, visualization and traffic analysis. CAIDA also developed popular measurement tools, such as NeTraMet or CoralReef [8, 62], and recently founded the Internet measurement data catalog DatCat [53]. Furthermore, CAIDA shares different datasets with the research community, such as security related data traces from their network telescope and some old packet-level header traces from US peering points. A number of relevant studies to passive Internet measurement and traffic analysis have been published throughout the years [63], some of them being real inspirations for the contributions in this thesis. These publications include the transport layer identification of P2P traffic by Karagiannis et al. [18, 64], the analyses of passively collected Internet traffic by Fomenkov et al. [65] and McCreary and Claffy [66] or the observations on fragmented traffic by Shannon et al. [67].

NLANR PMA

The Passive Measurement and Analysis Project (PMA) [68] of the National Laboratory for Applied Network Research (NLANR) ended officially in 2006, when CAIDA took over operational stewardship for NLANR machines and data, since both have been located at the San Diego Super Computer Center. The goal of NLANR PMA was to gain better understanding of the operation and behavior of the Internet by studying passive header traces. The traces have been achieved by daily measurements at different backbone network locations across the USA with speeds of up to OC48 (2.5Gbit/s). The measurements have been performed by specially designed nodes, the OC3MON and OC48MON systems [69], which were based on Endace DAG4.2 cards. Later, the OC48MON system greatly influenced the design of the IPMON system of Sprint. NLANR PMA also made packet header traces publicly available, which lead to a number of analysis studies by other researcher based on NLANR PMA data, such as the early study of wide-area Internet traffic characterstics by Thompson et al. [70] and the comparative study of TCP option deployment by Pentikousis and Badr [71].

SPRINT ATL

In early 2000, Sprint's Advanced Technology Labs (Sprint ATL) started with the design and deployment of a passive monitoring Infrastructure, called IPMON [40]. The IPMON system consists of a number of measurement nodes, a central data repository and an analysis platform for offline analysis of the data. The measurement nodes are technically very similar to the OC48MON systems and are located at geographically distributed Points of Presence (POPs) in order to collect data on different peering and backbone links, with speeds up to

OC192 (10Gbit/s). As a result, IPMON was able to collect packet-level traces on about 30 bidirectional links in the US Sprint IP backbone. The resulting trace analysis carried out between 2000 and 2005 revealed general traffic characteristics such as utilization, protocol breakdown and packet size distribution [72]. Currently, Sprint's applied research group is focusing on next-generation wireless systems, data mining and security. The latter research topic includes development of a continuous monitoring platform for high-speed IP backbone links, CMON, the successor of IPMON. CMON [73] is intended to provide a continuous packet stream for detection of anomalies, unusual events and malicious activities.

WIDE project and MAWI

The Widely Integrated Distributed Environment (WIDE) project [74] was launched in 1988 in Japan and is made up of more than 100 loosely bound organizations from all over the world. The visionary goal of WIDE is to construct a dependable Internet *'that can used by people from all walks of life in any situation with a sense of security'*. WIDE research activities cover all different layers of the Internet, including activities such as flow measurements with sFlow/NetFlow and analysis of IPv6, DNS and BGP routing information. The 'Measurement and Analysis on the WIDE Internet' (MAWI) working group furthermore provides a traffic repository of data captured on the WIDE backbone [75], focusing mainly on DNS and IPv6 traffic measurements.

IST SCAMPI / IST LOBSTER

SCAMPI [76] was a two and a half year European project sponsored by the Information Society Technologies (IST) program of the European Commission, starting in April 2002. SCAMPI involved ten European partner organizations, with the goal to develop a scalable monitoring platform for the Internet in order to promote the use of monitoring tools for improving services and technology. The original project was succeeded by another IST project until summer 2007, the LOBSTER [77] project. LOBSTER continued the deployment of an European Traffic Monitoring Infrastructure based on distributed monitoring sensors capable of collecting on link speeds of up to 10Gbit/s. Besides the deployment of a monitoring infrastructure, LOBSTER developed a number of monitoring and visualization tools, such as Stager [78], a tool for aggregating and presenting network statistics. SCAMPI and LOBSTER were also actively involved in the development of the IPFIX flow format standard [33]. Furthermore, LOBSTER made a number of network traces including attack traffic available for download [79]. Other activities included development of a generic anonymization framework for network traffic [80], which was developed after revealing vulnerabilities in existing pseudonymization approaches [28].

Other related work

Besides these big measurement projects, other relevant studies based on passive network measurement have been carried out by various researchers. Even if the available datasets did not reflect behavior of large parts of the Internet, some results are very significant and relevant. Allman studied deployment of TCP options within traffic from one particular webserver in a one and a half year period [81]. Also Medina et al. used passive measurements of two weeks duration from a local webserver to present usage of specific TCP features [82].

Other contributions had possibilities to record campus wide traffic for network analysis purposes. Arlitt and Williamson took a year long packet-level trace on the 100Mbit/s Ethernet campus network at the University of Calgary in order to analyse TCP reset behavior [83]. Also Moore and Papagiannaki used packet-level data collected on a campus network based on Gbit-Ethernet to compare network application identification methods [84]. These measurements were taken with Nprobe, a passive measurement architecture to perform traffic capturing and processing at full line-rate without packet loss [85]. Finally, Mori et al. collected packet-level traces on the external 100 Mbit/s links of an University during a one month period to compare flow characteristics between WWW and P2P traffic [86].

Measurements from networks with higher aggregation are usually only available in form of flow data. Gerber et al. e.g. had access to ten months of flow level data collected on several broadband ISPs, which was used to quantify P2P traffic in the Internet during the year 2002. [87]. Perenyi et al. also based their identification and analysis method of peer to peer (P2P) traffic on NetFlow data from an ADSL network with around a thousand of ADSL subsribers in Hungaria [88].

List of References

[1] Allen Householder, Kevin Houle, and Chad Dougherty, "Computer attack trends challenge internet security," *Computer*, vol. 35, no. 4, pp. 5–7, 2002.

[2] Sally Floyd and Eddie Kohler, "Internet research needs better models," Princeton, NJ, USA, 2003, vol. 33 of *Comput. Commun. Rev. (USA)*, pp. 29–34, ACM.

[3] Steven McCanne and Van Jacobson, "The BSD packet filter: A new architecture for user-level packet capture," in *USENIX Winter*, 1993, pp. 259–270.

[4] Endace, "Dag network monitoring cards," 2007, `http://www.endace.com/our-products/dag-network-monitoring-cards/`.

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

[6] J.D. Case, M. Fedor, M.L. Schoffstall, and J. Davin, "Simple Network Management Protocol (SNMP)," RFC 1157 (Historic), 1990.

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

[8] Ken Keys, David Moore, Ryan Koga, Edouard Lagache, Michael Tesch, and k claffy, "The architecture of CoralReef: an Internet traffic monitoring software suite," in *A workshop on Passive and Active Measurements, PAM '01*, 2001.

[9] "Directive 95/46/ec of the european parilament and of the council," 1995, `http://ec.europa.eu/justice_home/fsj/privacy/docs/95-46-ce/dir1995-46_p%art1_en.pdf`.

[10] "Directive 2002/58/ec of the european parilament and of the council," 2002, `http://eur-lex.europa.eu/LexUriServ/site/en/oj/2002/l_201/l_20120020731%en00370047.pdf`.

[11] "Directive 2006/24/ec of the european parilament and of the council," 2006, `http://eur-lex.europa.eu/LexUriServ/site/en/oj/2006/l_105/l_10520060413%en00540063.pdf`.

[12] Douglas C. Sicker, Paul Ohm, and Dirk 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.

[13] "18 united states code §2511," `http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00002511---%-000-.html`.

[14] "18 united states code §3127," `http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00003127---%-000-.html`.

[15] "18 united states code §2701," `http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00002701---%-000-.html`.

[16] "18 united states code §2702," `http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00002702---%-000-.html`.

[17] "18 united states code §2703," `http://www4.law.cornell.edu/uscode/html/uscode18/usc_sec_18_00002703---%-000-.html`.

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

[19] S. Coull, C. Wright, F. Monrose, M. Collins, and 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.

[20] Ruoming Pang, Mark Allman, Vern Paxson, and Jason Lee, "The devil and packet trace anonymization," *SIGCOMM Comput. Commun. Rev.*, vol. 36, no. 1, pp. 29–38, 2006.

[21] Jun Xu, Jinliang Fan, Mostafa H. Ammar, and Sue 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.

[22] Tatu 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`.

[23] Tadayoshi Kohno, Andre Broido, and K. C. Claffy, "Remote physical device fingerprinting," *IEEE Trans. Dependable Secur. Comput.*, vol. 2, no. 2, pp. 93–108, 2005.

[24] Mark Alllman and Vern 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.

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

[26] A. Slagell, J. Wang, and 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.

[27] Ramaswamy Ramaswamy, Ning Weng, and Tilman Wolf, "An ixa-based network measurement node," in *Proceedings of Intel IXA University Summit*, Hudson, MA, USA, 2004.

[28] Tønnes Brekne and André Årnes, "Circumventing ip-address pseudonymization," in *Proceedings of the Third IASTED International Conference on Communications and Computer Networks*, Marina del Rey, CA, USA, 2005.

[29] V. Paxson, "Growth trends in wide-area tcp connections," *Network, IEEE*, vol. 8, no. 4, pp. 8–17, Jul/Aug 1994.

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

[31] Baek-Young Choi, Jaesung Park, and Zhi-Li Zhang, "Adaptive packet sampling for accurate and scalable flow measurement," *Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE*, vol. 3, pp. 1448–1452 Vol.3, 29 Nov.-3 Dec. 2004.

[32] T.Zseby, M. Molina, N.Duffield, S.Niccolini, and 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%`.

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

[34] Cristian Estan and George 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.

[35] Nick Duffield, Carsten Lund, and Mikkel 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.

[36] Edith Cohen, Nick Duffield, Haim Kaplan, Carsten Lund, and Mikkel 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.

[37] Miquel Carsi Caballer and Lei Zhan, "Compression of internet header traces," Tech. Rep., Master Thesis, Chalmers University of Technology, Department of Computer Science and Engineering, 2006.

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

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

[40] Chuck Fraleigh, Sue Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and Christophe Diot, "Packet-level traffic measurements from the sprint ip backbone," *IEEE Network*, vol. 17, no. 6, pp. 6–16, 2003.

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

[42] Vern 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.

[43] Jörg Micheel, Stephen Donnelly, and Ian Graham, "Precision timestamping of network packets," in *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, 2001, pp. 273–277.

[44] Attila Pásztor and Darryl 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.

[45] S. Donnelly, "Endace dag timestamping whitepaper," 2006, Endace, `http://www.endace.com/`.

[46] EndRun Technologies, "CDMA Network Time Server," 2007, Datasheet, `http://www.endruntechnologies.com/pdf/TempusLxCDMA.pdf`.

[47] Per Olof Hedekvist, Ragne Emardson, Sven-Christian Ebenhag, and Kenneth Jaldehag, "Utilizing an active fiber optic communication network for accurate time distribution," *Transparent Optical Networks, 2007. ICTON '07. 9th International Conference on*, vol. 1, pp. 50–53, 1-5 July 2007.

[48] S.B. Moon, P. Skelly, and 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*, pp. 227–234, 1999.

[49] Li Zhang, Zhen Liu, and 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*, pp. 160–169, 2002.

[50] Yu Lin, GengSheng Kuo, Hongbo Wang, Shiduan Cheng, and Shihong 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*, vol. 3, pp. 1425–1430 Vol.3, 2004.

[51] Junfeng Wang, Mingtian Zhou, and Hongxia Zhou, "Clock synchronization for internet measurements: a clustering algorithm," *Comput. Networks*, vol. 45, no. 6, pp. 731–741, 2004.

[52] Hechmi Khlifi and Jean-Charles Grégoire, "Low-complexity offline and online clock skew estimation and removal," *Comput. Networks*, vol. 50, no. 11, pp. 1872–1884, 2006.

[53] Colleen Shannon, David Moore, Ken Keys, Marina Fomenkov, Bradley Huffaker, and k claffy, "The internet measurement data catalog," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 5, pp. 97–100, 2005.

[54] Richard Rockwell and Ronald Abeles, "Guest editorial: Sharing and archiving data is fundamental to scientific progress," *The Journals of Gerontology: Series B Psychological sciences and social sciences*, vol. 53B, pp. 5–8, 1998.

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

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

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

[58] "Wand network research group," `http://www.wand.net.nz/`.

[59] "Wits: Waikato internet traffic storage," `http://www.wand.net.nz/wits/`.

[60] Richard Nelson, Daniel Lawson, and Perry Lorier, "Analysis of long duration traces," *SIGCOMM Comput. Commun. Rev.*, vol. 35, no. 1, pp. 45–52, 2005.

[61] "The caida web site," `http://www.caida.org/`.

[62] "The caida tools site," `http://www.caida.org/tools/`.

[63] "List of papers by caida," `http://www.caida.org/publications/papers/bytopic/`.

[64] Thomas Karagiannis, Andre Broido, Michalis Faloutsos, and Kc Claffy, "Transport layer identification of p2p traffic," in *Proceedings of the 4th ACM Conference on Internet Measurement*, Taormina, Sicily, Italy, 2004.

[65] Marina Fomenkov, Ken Keys, David Moore, and K Claffy, "Longitudinal study of internet traffic in 1998-2003," in *WISICT '04: Proceedings of the winter international synposium on Information and communication technologies*, 2004.

[66] Sean McCreary and KC Claffy, "Trends in wide area ip traffic patterns - a view from ames internet exchange," Tech. Rep., CAIDA, San Diego Supercomputer Center, 2000.

[67] Colleen Shannon, David Moore, and KC Claffy, "Beyond folklore: observations on fragmented traffic," *IEEE/ACM Transactions on Networking*, vol. 10, no. 6, pp. 709–20, 2002.

[68] "Nlanr passive measurement and analysis project," `http://pma.nlanr.net/`.

[69] Joel Apisdorf, K. Claffy, Kevin Thompson, and Rick Wilder, "Oc3mon: Flexible, affordable, high performance staistics collection," in *LISA '96: Proceedings of the 10th USENIX conference on System administration*, Berkeley, CA, USA, 1996.

[70] Kevin Thompson, Gregory J. Miller, and Rick Wilder, "Wide-area internet traffic patterns and characteristics," *IEEE Network*, vol. 11, no. 6, pp. 10–23, 1997.

[71] Kostas Pentikousis and Hussein Badr, "Quantifying the deployment of tcp options - a comparative study," *IEEE Communications Letters*, vol. 8, no. 10, pp. 647–9, 2004.

[72] "Sprint ip data analysis trace collection overview," `http://ipmon.sprint.com/packstat/packetoverview.php`.

[73] K. To, T. Ye, and S. Bhattacharyya, "Cmon: A general-purpose continuous ip backbone traffic analysis platform," Tech. Rep., Sprint ATL, 2004, Research Report RR04-ATL-110309.

[74] "The wide project," `http://www.wide.ad.jp/`.

[75] "Packet traces from wide backbone," `http://tracer.csl.sony.co.jp/mawi/`.

[76] "The ist scampi project," `http://www.ist-scampi.org/`.

[77] "The ist lobster project," `http://www.ist-lobster.org/`.

[78] "The stager visualization package," `http://software.uninett.no/stager/`.

[79] "Lobster attack traces," `http://lobster.ics.forth.gr/traces/`.

[80] D. Koukis, S. Antonatos, D. Antoniades, E.P. Markatos, and P. Trimintzios, "A generic anonymization framework for network traffic," *Communications, 2006. ICC '06. IEEE International Conference on*, vol. 5, June 2006.

[81] Mark Allman, "A web server's view of the transport layer," *SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 5, 2000.

[82] Alberto Medina, Mark Allman, and Sally Floyd, "Measuring the evolution of transport protocols in the internet," *Computer Communication Review*, vol. 35, no. 2, pp. 37–51, 2005.

[83] M. Arlitt and C. Williamson, "An analysis of tcp reset behaviour on the internet," *Computer Communication Review*, vol. 35, no. 1, pp. 37–44, 2005.

[84] Andrew Moore and Konstantina Papagiannaki, "Toward the Accurate Identification of Network Applications," in *Proceedings of the Passive and Active Measurement Workshop (PAM2005)*, 2005.

[85] Andrew Moore, James Hall, Christian Kreibich, Euan Harris, and Ian Pratt, "Architecture of a Network Monitor," in *Passive and Active Measurement Workshop 2003 (PAM2003)*, 2003.

[86] Tatsuya Mori, Masato Uchida, and Shigeki Goto, "Flow analysis of internet traffic: World wide web versus peer-to-peer," *Systems and Computers in Japan*, vol. 36, no. 11, pp. 70–81, 2005.

[87] Alexandre Gerber, Joseph Houle, Han Nguyen, Matthew Roughan, and Subhabrata Sen, "P2p the gorilla in the cable," Chicago, IL, USA, 2003, NCTA National Show.

[88] M. Perenyi, Dang Trang Dinh, A. Gefferth, and S. Molnar, "Identification and analysis of peer-to-peer traffic," *Journal of Communications*, vol. 1, no. 7, pp. 36–46, 2006.

Part II

THE MONNET PROJECT

5 The MonNet project

This chapter provides a description of the MonNet project, a project for passive Internet traffic measurement and analysis. After giving some project background, including a description of the measurement location in SUNET (the Swedish University Computer Network) and some preparatory tasks (Section 5.1), the technical solution of the MonNet measurement infrastructure is presented by describing the measurement nodes and the processing platform (Section 5.2). Next, the pre-processing and analysis procedures of the resulting packet-level traces are described in Sections 5.3 and 5.4. Pre-processing steps include sanitization and de-sensitization of the traces, while actual analysis procedures range from packet-level and flow-level processing to traffic classification. Section 5.5 will then summarize the scientific results of the analyses, before an outlook on future research possibilities is finally given in Section 5.6.

5.1 Project background

Besides a presentation of the topology of the network measured, this section describes some preparatory steps which have been carried out before the actual measurements on the SUNET backbone links could be performed.

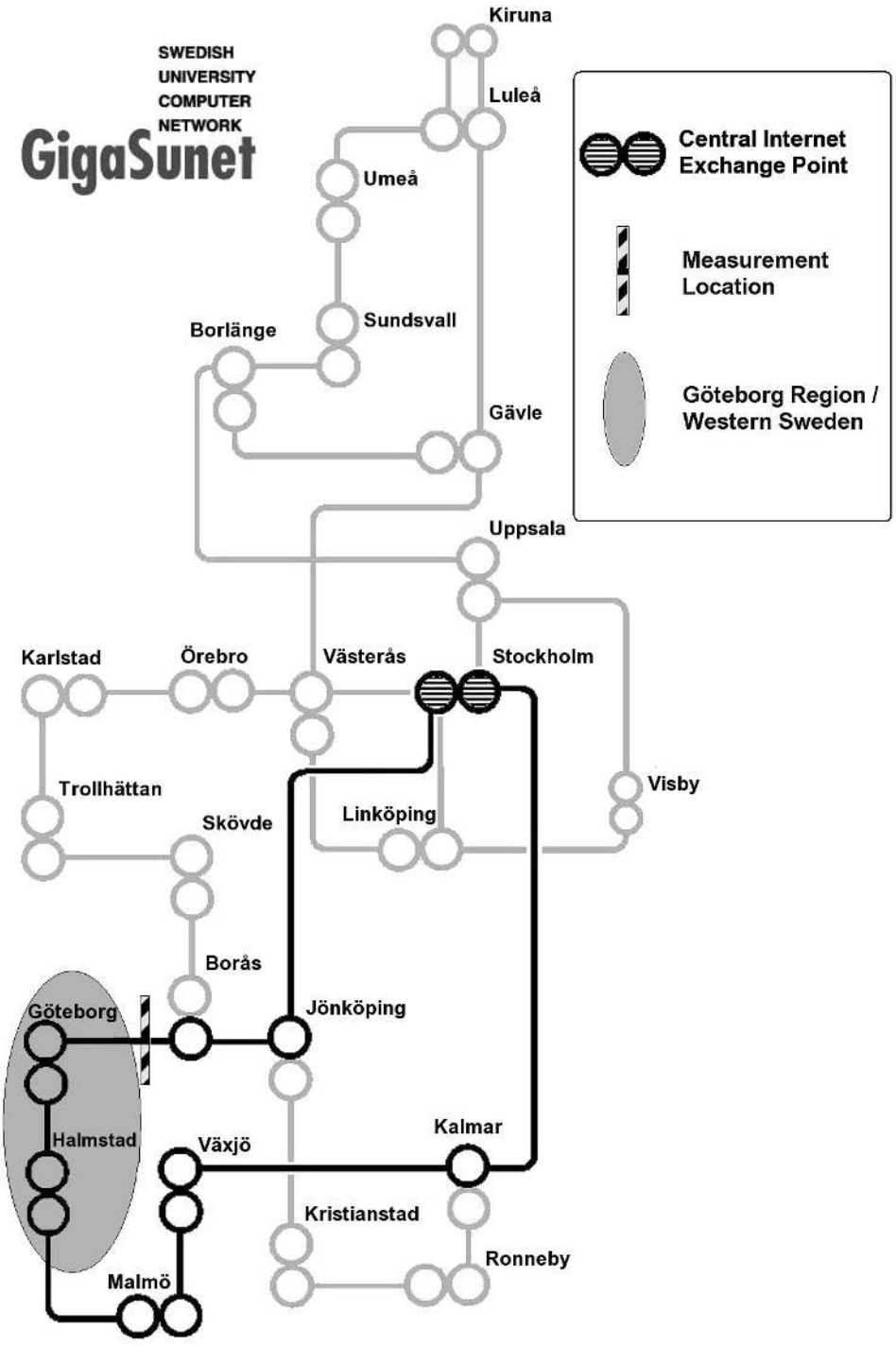


Figure 5.1: *Internal GigaSUNET topology with network ring measured (black lines)*

5.1.1 Description of the measured network

The measurement traces analyzed in this thesis have been collected on the previous generation of the SUNET backbone network, called GigaSUNET. GigaSUNET was officially in operation until January 2007, when it was replaced by the current generation, called Opto-SUNET. The GigaSUNET backbone consisted of four core rings joining together at a central Internet exchange point in Stockholm. Each ring used Cisco OC192 PoS technology over DWDM channels to interconnect all POPs, i.e. all University cities in Sweden. The topology of the internal GigaSUNET backbone is illustrated in Fig. 5.1. POPs are displayed with two circles in order to indicate the two core routers connecting that POP with the ones in the neighboring cities. Core routers are furthermore connected to an access network within the region, providing access to the SUNET backbone for regional customers, such as Universities and student networks. The OC192 links connecting POPs are illustrated as grey lines, with exception of the ring on which the measurements have been performed, which is colored in black. The traffic traces have been collected on the links between the cities of Göteborg and Borås, on the outermost part of the ring. This means that traffic passing the ring between the region of Göteborg (the grey shaded area) and the main Internet (peering with SUNET in Stockholm) was primarily routed via the tapped links, taking Borås as the next hop. This behavior was confirmed by SNMP statistics, showing that traffic amounts between Göteborg and Borås have been an order of magnitude larger than the amounts of traffic transfered between Halmstad and Malmö.

On the measurement location between Göteborg and Borås, backbone traffic was collected on two OC192 (10Gbit/s) links, one for each direction. The links have been tapped between the core router in direction of Borås and the DWDM system, connected to the 10Gbit/s channels leased from an operator. The two links measured provide the Internet backbone for two major Universities, a substantial number of student dormitories and a number of smaller Universities and research Institutes. Furthermore, around 14% of the collected traffic is exchange traffic with a local access point in Göteborg, providing peering between regional ISPs and SUNET as illustrated in Fig. 5.2. Thus, a significant part of the traffic is transit traffic. Due to hot-potato routing, transit traffic is in many cases routed asymmetrically, which means that around 10% of the measured connections exhibit asymmetrical properties. Summarized, the resulting traffic traces constitute a medium level of aggregation, between campus-wide traffic and tier-1 backbone traffic. We believe that this type of network, with smaller local exchange points, represents an upcoming class of networks.

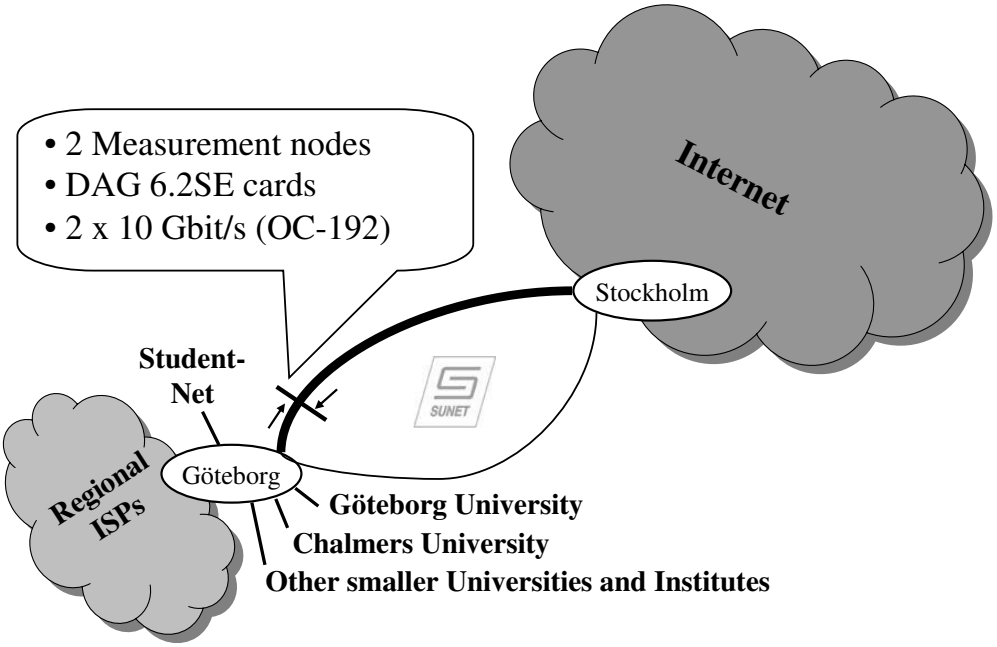


Figure 5.2: *Illustration of the measurement location between Göteborg and the main Internet*

5.1.2 Preparatory tasks and project administration

Before network measurements can be started, a number of preparatory steps need to be performed. First, MonNet, as a project regarding Internet and SUNET traffic measurements and analysis, was proposed to the SUNET 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 Göteborg. 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.

As a next step the measurement location was chosen as described above. This choice was in the first place made to be able to obtain traces of data transfered between a regional network and the main Internet. The chosen location has the additional feature of lying in the same city as the research group, located at the Chalmers University in central Göteborg. 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 un-

expected 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 [1] and Sprint's IPMON [2], provided valuable inspiration. The resulting technical solution will be described in detail in the next section.

Even if the preparatory tasks could be listed here very briefly, it is worth mentioning that they turned out the 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.

5.2 Technical solution

In the following paragraphs, the hardware used for the measurement and analysis infrastructure is described in detail. This includes *two measurement nodes* and one additional *processing platform*, the latter being used as storage, analysis platform and database for network traces. Optical splitters have been used to tap the two OC192 links, one for each direction. The splitters have been attached to two measurement nodes on-site, which also performed the pre-processing of the traces, as described later in Section 5.3. Traces have always been collected simultaneously for both directions. For final analysis, the network traces have been transferred to the processing platform at the Division of Computer Engineering at Chalmers University.

5.2.1 Measurement nodes

The two measurement nodes are designed and configured identically. 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. The cards are capable of collecting data on PoS and 10Gbit-Ethernet links with speeds of up to 10Gbit/s. The DAG cards have been configured with a buffer of 512MB reserved from the main memory. Furthermore, the DAG cards are configured to capture the first 120 bytes of each PoS frame to ensure that

the entire network and transport header information is preserved. The remaining payload fractions have been removed during the pre-processing of the traces (Section 5.3). The average packet size on the links measured lies around 700 bytes, which means a maximal throughput of around 1.8 million frames per second on a 10Gbit/s link. Since 44% of all frames are smaller than 120 bytes and therefore kept un-truncated, the truncated packet fragments stored have an average size of about 88 bytes, assuming even distribution of packet sizes. This means that at maximum link utilization of 10Gbit/s, about 160MByte/s need to be transfered to disk after truncating packet larger than 120 bytes, which is done online by the DAG card. However, due to heavy over-provisioning of the links measured, the nodes rarely needed to store more than 20MByte/s on disk during the MonNet measurement campaigns. Occasional traffic spikes reach of course much higher throughput values on short time scales, but these short spikes can be buffered in the reserved main memory, given sufficient I/O bus performance.

A measurement node consists of two AMD Opteron 64-bit processors with 2 GHz clock frequency and a total of 2GB of main memory, 1GB for each processor. Besides one system disk connected to the IDE controller, 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 411GB of cumulated disk-space for preliminary storage of collected network traces. The nominal throughput of the dual-channel Ultra-320 SCSI interface is 2x320MByte/s, which is sufficiently fast for the maximum collection speed of 160MByte/s with packet truncation at 120 bytes. The DAG cards and the SCSI controller are connected to CPU and main memory via two independent 64-bit PCI-X buses, operating with 133MHz. The PCI-X buses provide a nominal speed of 1000MByte/s, which is again sufficient for the chosen configuration. However, the throughput limits of the storage system and I/O buses would be a bottleneck in case of complete-packet capturing on full line speed of OC192 with 1250MByte/s.

During measurements, the two DAG cards have been synchronized to each other using Endace's DUCK Time Synchronization [3, 4] with no external reference time. Before and after measurements, the DAG cards were synchronized to reference time 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 have been reported earlier by Paxson in [5]. DUCK however can provide an accurate and consistent timestamping between the connected DAG cards ranging between $\pm$ 30ns according to Endace [4], even though their time might not be accurate with respect to true time (UTC). The tight synchronization between the measurements of opposing traffic directions allows simple merging of the unidirectional data into bidirectional traces.

5.2.2 Processing platform

After data collection and completion of the pre-processing procedures on the measurement nodes, the resulting traces have been transfered via a Gbit-Ethernet interface and a 2.5Git/s Internet connection to the storage and processing server located in the secured server room of the division of Computer Engineering at Chalmers University using secure copy (SCP). The processing platform is based on an Intel Xeon dual core CPU with 3.20GHz clock frequency and 2GB of main memory. An external SCSI array box with RAID5 configuration is attached to this platform, providing 2TB of storage. Besides storage of packet-level traces, the processing platform with the external storage is also housing a MYSQL database system, which is used for organizing results obtained by the different analyses of the raw traces, as described in Section 5.4.

5.3 Trace pre-processing

After storing the truncated data packets on the disks of the measurement nodes, the traces have been de-sensitized and sanitized in offline fashion, since online pre-processing in real-time is unfeasable due to computational limitations. De-sensitization and sanitization have therefore been carried out by batch jobs immediately after collection of the traces, in order to minimize the storage time of unprocessed and privacy-sensitive network traces. The pre-processing steps are described in the following paragraphs, together with some summarizing facts about the resulting network traces.

5.3.1 Trace de-sensitization

By trace de-sensitization the removing of all sensitive information to ensure privacy and confidentiality is meant. As a first step, the remaining payload beyond transport layer was removed using CAIDA's *CoralReef* [6] *crl_to_dag* utility. As a next step, IP addresses in the IPv4 headers have been anonymized using a customized anonymizer program based on the prefix-preserving *CryptoPAN* [7]. 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 of the measurements. This encryption key is kept safe and used for anonymization on the measurement nodes only.

5.3.2 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 or 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 delete upon completion of the pre-processing procedures, and only de-sensitized and sanitized versions of the traces have been kept. If errors have been detected, the pre-processing procedure has been stopped and further steps have been postponed, requesting manual inspection. Traces with major errors have consequently been deleted entirely, whereas minor problems, such as single checksum inconsistencies, have been documented in the meta-data and the pre-processing procedure was continued with the remaining steps. The sanitization procedure included the following checks:

- are timestamps strictly monotonic 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? (no packet arrival rate of zero packet/s)
- are there any occurrences of identical IP headers within consecutive frames?
- are there any IP header checksum errors?
- are all recorded frames of known type (i.e. POS with HDLC framing)?
- have records been lost during transfer to main memory (e.g. due to I/O bus limits)?
- have records been further truncated due to insufficient buffer space?
- have there been any receiver errors (i.e. link errors, such as incorrect light levels on the fiber and HDLC FCS (CRC) errors)?
- have there been any other internal errors on the DAG card?

As a result of the de-sensitization process, a small number of traces has been discarded due to major measurement errors. The corresponding traces in the opposite directions have in these cases been deleted as well. Furthermore, infrequently the DAG cards discarded single frames due to receiver errors, typically HDLC CRC errors. Some frames have also been reported as corrupted by the sanitization process due to IP checksum errors. Since the HDLC CRC was shown to be correct, this could be cases where IP checksum and CRC disagree [8]. 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 [9]. Since such missing or corrupted packets are very small in number, the traces have still been used for analysis, but missing packets and IP checksum errors have been documented in the attached meta-data file.

5.3.3 Resulting datasets

MonNet data traces have been recorded in two measurement campaigns during 2006. Datasets have been collected in April (spring dataset) and in the time from September to November 2006 (fall dataset) on the above described measurement location on SUNET. At each measurement time, traces have been stored simultaneously for both directions on the two measurement nodes. In spring, four traces of 20 minutes duration have been collected each day at identical times (2AM, 10AM, 2PM, 8PM) during a period of 20 days. The times have been chosen to cover business, non-business and nighttime hours. The fall dataset was collected on the same location at 277 randomized times during 80 days in fall 2006. At each random time, a trace of 10 minutes duration was stored. Randomized times have been chosen in order to provide a good statistical representation of Internet traffic characteristics at the specific time-period and location. A thorough documentation of both datasets and each individual trace can be found on DatCat, the Internet Measurement Data Catalog [10].

The collection process and the different pre-processing steps have been well documented for each single trace. The resulting meta-data was stored in a file together with a checksum digest of the particular trace, in order provide distinctive association. Meta-data includes a short description of the measurement location, direction of the measured link, timing information, status information of the DAG card and results of the three trace sanitization passes (before payload removal, after payload removal and before anonymization, after anonymization). Thereby, the meta-data provides a summary about errors detected, which includes counts of occasionally observed receiver errors (HDLC CRC errors) and the exact frame positions of frames including IP header checksum errors.

In total, the resulting datasets represent 71 hours of backbone traffic, collected on 106 days. The traces include 39 billion IPv4 packets, carrying 27 TB of data. The traces contain mainly IPv4 packets (99.97%). The remaining traffic consists of IPv6 BGP Multicast messages, CLNP routing updates (IS-IS) and Cisco Discovery Protocol (CDP) messages. Furthermore, around 40 currently unidentified frames have been observed each minute. These frames seem to have random address and control bytes in their Cisco HDLC headers, with non-standard ethertypes of 0x4000 or 0x0000. However, the purpose of these frames is still unclear.

The 148 traces collected at 74 different times in April (April dataset [11]) include 10.7 billion IPv4 packets, carrying 7.6 TB of data. During single 20 minute intervals, between 16.000 and 35.000 unique IP addresses have been observed inside the region of western Sweden connecting to 370.000-820.000 unique IP addresses in the rest of the Internet.

The fall dataset [12] consists of 554 traces collected at 277 measurement times during 80 days from September to November 2006 include 27.9 billion IPv4 packets, carrying 19.5 TB of data. During single 10 minute intervals, between 13.000 and 37.000 unique IP addresses have been observed inside the region of western Sweden connecting to 300.000-1.000.000 unique IP addresses outside (i.e. in the rest of the Internet).

5.4 Analysis approaches

So far, only the measurement processes including data pre-processing have been discussed. In this section, the analysis approaches used to extract scientific results as presented in Papers I-IV are outlined. The three presented analysis methodologies, packet-level analysis, flow-level analysis and traffic classification, are partly depending on each other.

Packet-level analysis

After the de-sensitized and sanitized traces have been stored on the processing platform, a rather straight-forward packet-level analysis has been conducted. An analysis program was run on each trace 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 or anomaly. The resulting database consists of tables for specificly interesting features, such as IP header length, IP packet length, TCP options and different kinds of anomalous behavior. In the tables data was summarized per direction and per measurement interval (i.e. trace time), which allowed analysis of the data in different dimension by issuing respective SQL queries. The results of the packet-level analysis are summarized in **Paper I** and to some extent in Sections 3 and 4 of **Paper II**.

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 been summarized to flows by the use of 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, with each row representing a summary of exactly one flow or connection.

A flow is defined by the traditional 5-tuple of source and destination IP and port numbers and the transport protocol. As transport protocols, only TCP and UDP have been considered. TCP flows represent connections, and are therefore further separated by SYN, FIN and RST packets. Additional SYN segments for a specific tuple can sometimes be seen in the same direction within short time intervals, as usually the case within scanning campaigns. In such cases, further 'connections' have been opened within the analysis program in order to record the pure SYN packets separately. Following non-SYN packets are then always added to the most recently opened connection of the particular tuple. Since UDP offers no connection establishment or termination, UDP flows have been defined as the sum of bidirectional packets observed between a specific 5-tuple during a specified time interval. For **Paper II**, this timeout was specified as 20 minutes, taking advantage of the measurement duration of the traces in the April dataset. For **Paper III** and **Paper IV**, UDP flows are separated by the commonly accepted timeout of 64 seconds [13], in order to obtain results comparable to related work [14].

Traffic classification

Traffic classification on application level was done based on a set of heuristics regarding connection patterns of individual endpoints in the Internet. A detailed description and verification of the heuristics can be found in **Paper III**. The traffic classification was almost entirely done by complex SQL statements within the database, starting with flow tables as generated during the flow-level analysis. The heuristics have been applied to the flow tables in 10 minute intervals, which means that every interval is analyzed self-contained, without memory of previous intervals. Most of the 15 heuristics (with two exceptions) have first been applied independently to all flows. For each flow, a bit-mask was set in a separate table in the database according to matching rules. This approach makes it possible to verify each heuristic separately and to investigate the effects of different priority rankings of the heuristics. After empirical exploration of the most suitable prioritization scheme for the heuristics, an additional bit mask associated with each flow was set, indicating the final traffic classification into classes such as Web, P2P and attack traffic. The original flow tables together with the associated classification tables allow a convenient way to analyse and compare flow and connection characteristics among traffic of different network applications, as successfully done for **Paper IV**.

5.5 Scientific contribution

After providing a general introduction to the topic of Internet measurement, including some basic guidelines for measurement activities based on experiences and lessons learned from a passive measurement project (Chapter 3), and a detailed description of a successful project for passive measurement on an Internet backbone (Chapter 5), this section will summarize the scientific contribution of this thesis. The results provide a basic foundation for a better understanding of the Internet, and thereby form e.g. valuable input for refinement of future simulation models and improvement of security and intrusion detection systems. The research results have been reported in four papers, all of them presented and published at recognized scientific conferences. The following paragraphs outline the research progress, including pointers to the specific paper in bold font.

Following the collection of the data traces on SUNETs 10Gbit/s links, an initial packet-level analysis was intended to investigate the deployment of protocol specific features and accompanying anomalies of common Internet protocols (IP, TCP, UDP) on per-packet basis. Since these protocols allow some flexibility in implementation, including a variety of optional features, this type of statistics is important to support research and further development of these protocols. The results of **Paper I** consequently reflect the current characteristics of Internet backbone traffic by providing a comprehensive summary about protocol usage and highlighting misbehaviors and potential problems. However, the results presented not only provide contemporary traffic characteristics, they also raise new questions, such as unexplained differences between inbound and outbound traffic statistics.

In **Paper II**, the observed packets have therefore been correlated in order to better understand the features of today's network traffic. The more sophisticated analysis on flow-level revealed that the significant differences between backbone traffic going to and coming from the main Internet stem from differences in traffic composition - incoming links carry much more malicious traffic (like network scans), but also P2P traffic turned out to have large influence on traffic behavior. Besides showing that traffic is not necessarily as symmetrically shaped as it seems in high-level statistical summaries, the results finally also confirm the popular assumption that the Internet is rather hostile and unfriendly compared to typical University campus and student networks.

Since traffic classes, such as P2P and scanning traffic, are shown to have strong influence on the overall traffic characteristics, it was necessary to apply some kind of traffic classification on application level on the data. Initially, it was planned to use an existing and verified

classification technique. Since the datasets do not include packet payload and accurate training data is missing, payload signature methods [15] and classification based on statistical fingerprinting [16] were no options. Thus it was decided that applying connection pattern heuristics is the most suitable solution for this task. Previous work on connection pattern based classification [14, 17] is shown to yield significantly disagreeing results, including a substantial number of false positives when applied the MonNet datasets. As a result, in **Paper III** a refined and updated set of heuristics to classify backbone traffic according to network applications is presented. The heuristics do not require any packet payloads, but only take packet headers into account. This feature is not only relevant due to legal and privacy issues, it also allows classification of encrypted network data. The ability to classify encrypted traffic is gaining further relevance considering a recent study, which showed that an increasing number of the connections by popular P2P file sharing applications are already encrypted in modern networks [18]. The proposed classification method is intended to provide researchers and network operators with a simple and fast method to get insight into the type of data carried by their links. A complete application classification can be provided even for short 'snapshot' traces, including identification of P2P, Web and attack or malicious traffic. When applying the heuristics to the April dataset [11], in the best case only 0.2% of the data was left unclassified.

After being able to classify traffic, the three main traffic classes have been compared in terms of traffic volumes and signaling behavior in **Paper IV**. These traffic classes are (1) Web or HTTP traffic, including HTTPS; (2) P2P file sharing protocols, often based on overlay networks; (3) Malicious and attack traffic, i.e. network scans, sweeps and DoS attacks. Furthermore, the paper also highlights longitudinal trends and diurnal differences within each traffic class. It was shown that traffic volumes are increasing considerably, with P2P-traffic clearly dominating. In contrast, the amount of malicious and attack traffic remains constant, forming a constant 'background noise' in the Internet. P2P and Web traffic are shown to differ significantly in connection establishment and termination behavior. Additionally, an analysis of TCP option usage revealed that some options (e.g. Selective Acknowledgment, SACK) is still neglected by a number of popular web-servers, even though it is deployed by most web clients. The results of this study finally confirmed many of the assumptions made in the earlier papers regarding properties of isolated P2P and malicious traffic.

5.6 Future outlook

Due to the successes of the MonNet project during the last three years, the SUNET board granted a project extension. This opens up for deployment of the existing measurement system in the currently active OptoSUNET, in order to obtain new network traces. Since OptoSUNET has a completely new technology compared to GigaSUNET, a suitable measurement location was identified in Stockholm, on the peering links between SUNET and NORDUnet, the joint organization of Nordic national networks for research and education. This measurement location would make re-usage of the current MonNet equipment possible, and furthermore provide network data with even higher level of aggregation, since cumulated traffic from the entire SUNET could be captured. In order to expand future analysis possibilities, it would be desirable to record packet-header traces including data fragments beyond transport layer, such as HTTPS and SSH headers. While such extension of the records requires repeated legal discussions, it would allow investigation of new, interesting research questions, like deployment of SSL and SSH variants and their encryption methods.

It is also intended to intensify the cooperation with the Computer Security Group at Chalmers University in order to take advantage of their expertise in data logging and intrusion detection when analyzing the data gathered within the MonNet project. Such cooperation would offer new research possibilities, such as investigating the behavior of large-scale attacks, which go beyond the view of a local honeypots due to the high-level perspective of the backbone measurement infrastructure. Analysis of the highly aggregated backbone traffic can also be additional input for the design of intrusion detection systems, by performing a systematic, specification based analysis of protocol stack exploits as 'seen in the wild'.

Since the MonNet project does currently not posses equipment to measure on more than one location at a time, cooperation with related research groups all over the world could be beneficial for both parties. This would offer a new range of research possibilities, such as delay measurements or comparison of traffic characteristics on different locations at the same time. An interesting possibility in this respect is the *Day in the Life of the Internet* (DITL) project [19], a community-wide measurement experiment coordinated by CAIDA and WIDE. Even though the MonNet measurement infrastructure will most likely not be in place before the next DITL collection event in March 2008, participation in future events is an interesting and aspired possibility.

List of References

[1] Joel Apisdorf, K. Claffy, Kevin Thompson, and Rick Wilder, "Oc3mon: Flexible, affordable, high performance staistics collection," in *LISA '96: Proceedings of the 10th USENIX conference on System administration*, Berkeley, CA, USA, 1996.

[2] Chuck Fraleigh, Sue Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and Christophe Diot, "Packet-level traffic measurements from the sprint ip backbone," *IEEE Network*, vol. 17, no. 6, pp. 6–16, 2003.

[3] Jörg Micheel, Stephen Donnelly, and Ian Graham, "Precision timestamping of network packets," in *IMW '01: Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, 2001, pp. 273–277.

[4] S. Donnelly, "Endace dag timestamping whitepaper," 2006, Endace, `http://www.endace.com/`.

[5] Vern Paxson, "On calibrating measurements of packet transit times," *SIGMETRICS Perform. Eval. Rev.*, vol. 26, no. 1, pp. 11–21, 1998.

[6] Ken Keys, David Moore, Ryan Koga, Edouard Lagache, Michael Tesch, and k claffy, "The architecture of CoralReef: an Internet traffic monitoring software suite," in *A workshop on Passive and Active Measurements, PAM '01*, 2001.

[7] Jun Xu, Jinliang Fan, Mostafa H. Ammar, and Sue 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.

[8] Jonathan Stone and Craig Partridge, "When the crc and tcp checksum disagree," in *SIGCOMM '00: Proceedings of the Conference on Applications, Technology, Architecture and Protocols for Computer Communication*, 2000.

[9] A. Rijsinghani, "Computation of the Internet Checksum via Incremental Update," RFC 1624 (Informational), 1994.

[10] Wolfgang John and Sven Tafvelin, "SUNET OC 192 Traces (collection)," `http://imdc.datcat.org/collection/1-04L9-9=SUNET+OC+192+Traces` (accessed 080115).

[11] Wolfgang John and Sven Tafvelin, "SUNET OC 192 Traces, April 2006 (collection)," `http://imdc.datcat.org/collection/1-04HN-W=SUNET+OC+192+Traces\%2C+Apri%l+2006` (accessed 080115).

[12] Wolfgang John and Sven Tafvelin, "SUNET OC 192 Traces, fall 2006 (collection)," `http://imdc.datcat.org/collection/1-04HQ-3=SUNET+OC+192+Traces\%2C+fall%+2006` (accessed 080115).

[13] K.C. Claffy K.C., H.-W. Braun, and G.C. Polyzos, "A parameterizable methodology for internet traffic flow profiling," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 8, 1995.

[14] Thomas Karagiannis, Andre Broido, Michalis Faloutsos, and Kc Claffy, "Transport layer identification of p2p traffic," in *Proceedings of the 4th ACM Conference on Internet Measurement*, Taormina, Sicily, Italy, 2004.

[15] Subhabrata Sen, Oliver Spatscheck, and Dongmei Wang, "Accurate, scalable in-network identification of p2p traffic using application signatures," New York, USA, 2004, 13th International World Wide Web Conference.

[16] Manuel Crotti, Maurizio Dusi, Francesco Gringoli, and Luca Salgarelli, "Traffic classification through simple statistical fingerprinting," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 1, pp. 5–16, 2007, 1198257.

[17] M. Perenyi, Dang Trang Dinh, A. Gefferth, and S. Molnar, "Identification and analysis of peer-to-peer traffic," *Journal of Communications*, vol. 1, no. 7, pp. 36–46, 2006.

[18] Ipoque GmbH, "Internet study 2007," `http://www.ipoque.com/userfiles/file/Internet_study_2007_abstract_en.pd%f`.

[19] CAIDA and WIDE, "A day in the life of the internet," `http://www.caida.org/projects/ditl/`.

Part III

PAPERS

PAPER I

Wolfgang John and Sven Tafvelin

Analysis of Internet Backbone Traffic and Anomalies observed

IMC '07: Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement
San Diego, California, USA, 2007

Analysis of Internet Backbone Traffic and Header Anomalies Observed

Wolfgang John and Sven Tafvelin
Chalmers University of Technology
Email: {johnwolf,tafvelin}@chalmers.se

ABSTRACT

The dominating Internet protocols, IP and TCP, allow some flexibility in implementation, including a variety of optional features. To support research and further development of these protocols, it is crucial to know about current deployment of protocol specific features and accompanying anomalies. This work is intended to reflect the current characteristics of Internet backbone traffic and point out misbehaviors and potential problems. On 20 consecutive days in April 2006 bidirectional traffic was collected on an OC-192 backbone link. The analysis of the data provides a comprehensive summary about current protocol usage including comparisons to prior studies. Furthermore, header misbehaviors and anomalies were found within almost every aspect analyzed and are discussed in detail. These observations are important information for designers of network protocols, network application and network attack detection systems.[1]

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Network monitoring

General Terms

Measurement

Keywords

Internet Measurement, Traffic Analysis, Header Anomalies

1. INTRODUCTION

Today, the Internet has emerged as the key component for commercial and personal communication. One contributing factor to the still ongoing expansion of the Internet is its versatility and flexibility. Applications and protocols keep changing not only with time [1], but also within geographical locations. Unfortunately, this fast development has left little time or resources to integrate measurement and analysis possibilities into the Internet infrastructure. However, the Internet community needs to understand the nature of Internet traffic in order to support research and further development [2]. It is also important to know about current deployment of protocol specific features and possible misuse. This knowledge is especially relevant in order to improve the robustness of protocol implementations and network applications, since increasing bandwidth and growing numbers of Internet users also lead to increased misuse and anomalous behavior [3]. One way of acquiring better understanding is to measure and analyze real Internet traffic, preferably on highly aggregated links. The resulting comprehensive view is crucial for a better understanding of the applied technology and protocols and hence for the future development thereof. This is important for establishing simulation models [4] and will also bring up new insights for related research fields, such as network security or intrusion detection.

A number of studies on protocol specific features have been published earlier, based on a variety of datasets. Thompson et al. [5] presented wide-area Internet traffic characteristics on data recorded on OC-3 traffic monitors in 1997, including figures about packet size distribution and transport protocol decomposition. McCreary et al. [1] provided a longitudinal analysis of Internet traffic based on data collected on an OC3 link of the Ames Internet exchange in 1999 to 2000. Fractions of fragmented traffic were presented and the usage of Path MTU Discovery was inferred based on the packet size distribution. Shannon et al. [6] studied frequency and cause of IP fragmented traffic on data collected on different WANs (100Mbit Ethernet, OC3 and OC12) in March 2001. Fraleigh et al. [7] analyzed traffic measurements from the Sprint IP backbone, based on a number of traces taken on different OC12 and OC48 links in 2001-2002. Pentikousis et al. [8] indirectly quantified deployment of TCP options based on traces with incomplete header information. The data was collected between October 2003 and January 2004 on a number of OC3 and OC12 links by the NLANR/PMA. In that paper, recent figures about packet size distributions were presented as well. Already earlier, Allman [9] presented observations about usage of TCP options within traffic from a particular webserver in a one and a half year period from 1998-2000. Finally, in his investigations about the evolution of transport protocols, Medina et al. [10] presented usage of TCP features like ECN (RFC 3168) based on passive measurements on a local webserver during two weeks in February 2004.

[1]This work was supported by SUNET, the Swedish University Network

IMC'07, October 24–26, 2007, San Diego, California, USA.

Despite these existing studies, there is a need for further measurement studies [2, 11]. Continued analysis work needs to be done on updated real-world data in order to be able to follow trends and changes in network characteristics. Therefore, in this work we will consequently continue to analyze IP and TCP, as they are the most common protocols used in today's Internet, and compare the results to previous work. After describtion of the analyzed data in Section 2, we present our results for IP and TCP specific features in Section 3. Finally, Section 4 summarizes the main findings and draws conclusions.

2. METHODOLOGY

2.1 Collection of Traces

The traffic traces have been collected on the outermost part of an SDH ring running Packet over SONET (PoS). The traffic passing the ring to (outgoing) and from (incoming) the Internet is primarily routed via our tapped links. This expected behavior is confirmed by SNMP statistics showing a difference of almost an order of magnitude between the tapped link and the protection link. Simplified, we regard the measurements to be taken on links between the region of Göteborg, including exchange traffic with the regional access point, and the rest of the Internet.

On the two OC-192 links (two directions) we use optical splitters attached to two Endace DAG6.2SE cards. The DAG cards captured the first 120 bytes of each frame to ensure that the entire network and transport header information is preserved. The data collection was performed between the 7th of April 2006, 2AM and the 26th of April 2006, 10AM. During this period, we simultaneously for both directions collected four traces of 20 minutes each day at identical times. The times (2AM, 10AM, 2PM, 8PM) were chosen to cover business, non-business and nighttime hours. Due to measurement errors in one direction at four occasions we have excluded these traces and the corresponding traces in the opposite direction.

2.2 Processing and Analysis

After storing the data on disk, the payload beyond transport layer was removed and the traces were sanitized and desensitized. This was mainly done by using available tools like Endace's *dagtools* and CAIDA's *CoralReef*, accompanied by own tools for additional consistency checks, which have been applied after each preprocessing step to ensure sanity of the traces. 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. During our capturing sessions, the DAG cards discarded a total of 20 frames within 12 different traces due to receiver errors or HDLC CRC errors. Another 71 frames within 30 different traces had to be discarded after the sanitization process due to IP checksum errors.

By desensitization the removing of all sensitive information to ensure privacy and confidentiality is meant. The payload of the packets was removed earlier, so we finally anonymized IP addresses using the prefix preserving CryptoPAN [12]. After desensitization, the traces were moved to a central storage. An analysis program was run on the data to extract cummulated statistical data into a database. For packets of special interest, corresponding TCP flows have been extracted.

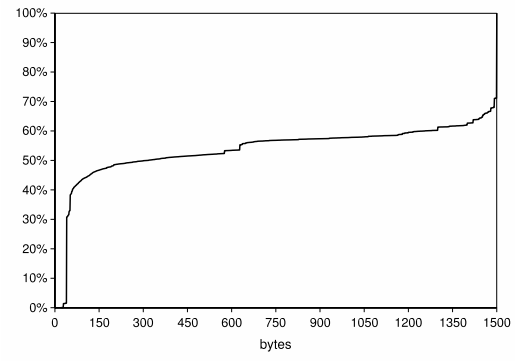


Figure 1: Cum. IPv4 Packet Size Distribution

3. RESULTS

The 148 traces analyzed sum up to 10.77 billion PoS frames, containing a total of 7.6 TB of data. 99.97% of the frames contain IPv4 packets, summing up to 99.99% of the carried data. The remaining traffic consists of different routing protocols (BGP, CLNP, CDP). The results in the remainder of this paper are based on IPv4 traffic only.

3.1 General Traffic Properties

3.1.1 IP packet size distribution

In earlier measurements, cumulative distribution of IPv4 packet lengths was reported to be trimodal, showing major modes at small packet sizes just above 40 bytes (TCP acknowledgments), large packets around 1500 bytes (Ethernet MTU) and default datagram sizes of 576 bytes according to RFC 879. Data collected between 1997 and 2002 reported about fractions of default datagram sizes from 10% up to 40% [5, 1, 6, 7]. Pentikousis et al. [8] however showed in 2004, that packet size distribution was no longer trimodal, but rather bimodal, with default datagram sizes accounting for only 3.8% of all packets.

Fig. 1 illustrates the cumulative distribution of IPv4 packet lengths in our traces of 2006. The distribution is still bimodal, with the major portion of lengths between 40 and 100 bytes and between 1400 and 1500 bytes (44% and 37% of all IPv4 packets, resp.). The usage of the default datagram size of 576 byte was further decreased to a fraction of only 0.95%, now not even being among the first three most significant modes anymore. This is caused by the predominance of Path MTU Discovery in today's TCP implementations, which is confirmed later by the analysis of the IP flags and the TCP maximum segment size (MSS) option. On the other hand, two other notable modes appeared at 628 bytes and 1300 bytes, representing 1.76% and 1.1% of the IPv4 traffic, resp.

An analysis of TCP flows including a lot of 628 byte packets showed that these packets typically appear after full sized packets (MSS of 1460), often with the PUSH flag set. We suspect that they are sent by applications doing 'TCP layer fragmentation' on 2KB blocks of data, indicating the end of data a data block by PUSH. This is confirmed by flows where smaller MSS values have been negotiated (e.g. 1452). In this cases, the following packets became larger (e.g. 636 bytes) to add up to 2048 bytes of payload again. Examples for applications using such 2KB blocks for data transfer can

be found in [13], where different file-sharing protocols using fixed block sizes are presented. A look at the TCP destination ports revealed that large fractions of this traffic are indeed sent to ports known to be used for popular file-sharing protocols like Bittorrent and DirectConnect. The notable step at 1300 bytes on the other hand could be explained by the recommended IP MTU for IPsec VPN tunnels [14].

Packets larger than 1500 bytes (Ethernet MTU) aggregate a fraction of 0.15%. Traffic of packets sized up to 8192 bytes was observed, but the major part (99.7%) accounts for packet sizes of 4470 bytes. A minor part of the >1500 byte sized packets represents BGP updates between backbone- or access routers. The majority of the large packet traffic (mainly 4470) could after thorough investigation be identified as customized data-transfer from a space observatory to a data center using jumbo-packets over Ethernet.

	2AM		10AM		2PM		8PM	
	Pkts	Data	Pkts	Data	Pkts	Data	Pkts	Data
TCP	91.3	97.6	91.5	96.8	93.2	97.1	91.4	97.2
UDP	8.5	2.3	7.6	2.8	6.1	2.7	8.3	2.7
ICMP	0.2	0.02	0.19	0.02	0.20	0.02	0.12	0.01
ESP	0.01	0.00	0.47	0.19	0.35	0.14	0.02	0.02
GRE	0.01	0.01	0.08	0.08	0.04	0.03	0.06	0.04

(a) IPv4 Protocol Breakdown (values in %)

OUTGOING UDP			
Date	Time	Packets	Data
2006-04-16	2PM	6.8	1.7
2006-04-16	8PM	40.6	5.1
2006-04-17	2AM	51.9	6.1
2006-04-17	10AM	58.1	7.1
2006-04-17	2PM	5.7	1.8

(b) UDP Burst (values in %)

Table 1: Transport Protocols

3.1.2 Transport protocols

The protocol breakdown in Table 1(a) once more confirms the dominance of TCP traffic. Compared to earlier measurements reporting about TCP accounting for around 90 - 95% of the data volume and for around 85-90% of IP packets, [5, 1, 6, 7], both fractions seem to be slightly larger in the analyzed SUNET data. In Table 1(a), the fractions of cumulated packets and bytes carried in the respective protocol are given in percent of total IPv4 traffic for the corresponding time.

An interesting observation can be made at the 2PM data. Here, the largest fraction of TCP and the lowest of UDP packets appear. A closer look at the differences between outgoing and incoming traffic revealed that three consecutive measurements on the outgoing link carried up to 58% UDP packets, not covering the 2PM traces, as shown in Table 1(b). These figures indicate a potential UDP burst of 14-24 hours of time. A detailed analysis showed that the packet length for the UDP packets causing the burst was just 29 bytes, leaving a single byte for UDP payload data. These packets were transmitted between a single sender and receiver address with varying port numbers. After reporting this network anomaly, the network support group of a University confirmed that the burst stemed from an UDP DoS script installed undetected on a webserver with a known vulnerability. Although TCP data was still predominant, a dominance of UDP packets over such a timespan could potentially lead to TCP starvation and raise serious concerns about Internet stability and fairness.

3.2 Analysis of IP Properties

3.2.1 IP type of service

The TOS field can optionally include codepoints for Explicit Congestion Notification (ECN) and Differentiated Services. 83.1% of the observed IPv4 packets store a value of zero in the TOS field, not applying the mechanisms above. Valid 'Pool 1' DiffServ Codepoints (RFC 2474) account for 16.8% of all TOS fields.

Medina et al. [10] reported about almost an doubling of ECN capable webservers from 1.1% in 2000 to 2.1% in 2004, but indicates that routers or middleboxes might erase ECT codepoints. In our data only 1.0 million IPv4 packets provide ECN capable transport (either one of the ECT bits set) and additionally 1.1 million packets actually show 'congestion experienced' (both bits set). This means that ECN is implemented in only around 0.02% of the IPv4 traffic. These numbers are consistent with the observations by Pentikousis et al. [8], suggesting that the number of ECN-aware routers is still very small.

3.2.2 IP Options

The analysis of IP options showed that they are virtually not used. Only 68 packets carrying IP options were observed. One 20-minute trace contained 45 packets with IP option 7 (Record Route) and 3 traces carried up to 12 packets with IP option 148 (Router Alert).

3.2.3 IP fragmentation

During the year 2000, McCreary et al. [1] observed an increase in the fraction IP packets carrying fragmented traffic from 0.03% to 0.15%. Indeed, one year later, Shannon et al. [6] reported fractions of fragmented traffic of up to 0.67%. Contrary to this trend, we found a much smaller fraction of 0.06% of fragmented traffic in the analyzed data. Even though these numbers are relatively small, there is still an order of magnitude difference between earlier and current results. 72% of the fragmented traffic in our data is transmitted during office hours, at 10AM and 2PM. We also observed that the amount of fragmented traffic on the incoming link is about 9 times higher than on the outgoing one.

While UDP and TCP are responsible for 97% and 3% respectively of all incoming fragmented segments, they just represent 19% and 18% of the outgoing. The remaining 63% of the outgoing fragmented traffic turned out to be IPsec ESP traffic (RFC 4303), observed between exactly one source and one receiver during working hours on weekdays. Each fragment series in this connection consists of one full length Ethernet MTU and one additional 72 byte fragment. This can easily be explained by an unsuitably configured host/VPN combination transmitting 1532 bytes (1572 - 40 bytes IP and TCP header) instead of the Ethernet MTU due to the additional ESP header. The dominance of UDP among fragmented traffic is not surprising, since Path MTU Discovery is a TCP feature only.

The first packets in all observed fragment series are in 96.7% sized larger or equal than 1300 bytes. This goes along with the assumption that fragments are sent in-order and the first segments should be full sized MTUs. It should be noted that 1.6% of first packets in fragment series are smaller than 576 bytes. This is not surprising, considering an earlier observation by Shannon et al. [6] that about 8% of fragment series are sent in reverse-order, sending the smallest segment first. This is accepted behavior, since the IP specification (RFC 791) does not prescribe any sizes of fragments. Another reason for small first segments are misconfigured networks or deliberate use of small MTUs, like serial links (RFC 1144) connected to the backbone. An example for such unusual small sized fragments of only 244 bytes will be given in the next subsection.

3.2.4 IP flags

The analysis of the IP flags (fragment bits) revealed that 91.3% of all observed IP packets have the don't fragment bit (DF) set, as proposed by Path MTU Discovery (RFC 1191). 8.65% use neither DF nor MF (more fragments) and 0.04% set solely the MF bit.

Following the IP specification (RFC 791) no other values are valid in the IP flag field. Nevertheless, we observed a total of 27,474 IPv4 packets from 70 distinct IP sources with DF and MF set simultaneously. About 35 of those invalid bit values are evenly observed among both directions in all traces, with exception of one burst of 21,768 packets in a trace of the incoming link. This burst stems from a 10 minutes long TCP flow between a local server on port 49999 and a remote client on the gaming port 1737 (UltimaD). Surprisingly, all the incoming traffic is fragmented to series of 244 byte long IP packets. The data carried by these fragment series adds up to full Ethernet MTUs size. Because being fragmented, each but the last fragment in a series has the MF bit set. Disregarding its actual fragmentation, each fragment also has the DF bit set. A similar behavior could be observed on the outgoing link, where one source generates 85% of all outgoing DF+MF packets, evenly distributed over 70 out of 76 measured times. Again, each IP packet has the DF bit set by default, while MF is set additionally when fragmentation is needed. Looking at the traffic pattern and considering that UDP port 53 is used, it seems to be obvious that there is a DNS server using improper protocol stacks inside the Göteborg region.

Additionally, we observed a total of 233 cases of a reserved bit with value 1, appearing in small numbers in most of the collected traces and stemming from 126 distinct sources. According to the IP standard (RFC 791) the reserved bit must be zero, so this behavior has to be regarded as misbehavior.

3.3 Analysis of TCP Properties

3.3.1 TCP Options

In an early study, Allman [9] reported about portions of hosts applying the Window Scale (WS) and Timestamp (TS) options, both increasing from about 15% to 20% during a 15 month period from 1998 to 2000. The SACK permitted option was shown to increase even further from 7% to 40%. No numbers for hosts applying the MMS option were given. The more recent approach to quantify TCP option deployment by Pentikousis et al. in 2004 [8] was unfortunately carried out on traces with incomplete header information. Since TCP option data was not available in these traces, their deployment had consequently to be analyzed indirectly. Our results, based on traces including complete header information, show that this indirect approach yielded quite accurate results.

Table 2(a) shows the deployment of the most important TCP options as fractions of the SYN and SYN/ACK segments, divided into summaries of the four times each day. The results show that MSS and SACK permitted options are widely used during connection establishment (on average 99.2% and 89.9% resp.). The positive trend of the SACK option deployment, as indicated by Allman, was obviously continued and the inferred values of Pentikousis et al. are finally confirmed. The frequent usage of the MSS option again indicates the dominance of Path MTU Discovery in TCP connections, since an advertised MSS is the precondition for this technique. The WS and TS options on the other hand are still applied to the same extent as in 2000 (17.9% and 14.5% resp.). In Table 2(b) the occurrence of

Kind	2AM	10AM	2PM	8PM
2(MSS)	99.0%	98.7%	99.7%	99.1%
3(WS)	21.4%	18.4%	16.6%	16.5%
4(SACK perm.)	91.0%	86.6%	88.9%	89.8%
8(TS)	18.2%	15.3%	13.3%	12.8%

(a) TCP Options in SYN segments

Kind	2AM	10AM	2PM	8PM
No Opt.	86.5%	85.2%	87.3%	88.6%
5(SACK)	3.1%	2.8%	2.9%	3.1%
8(TS)	9.7%	11.2%	9.0%	7.6%
19(MD5)	0.02%	0.02%	0.01%	0.01%

(b) TCP Options in all segments

Table 2: TCP Option Deployment

TCP options with respect to all TCP segments is summarized. Around 87% of the TCP segments do not carry any options at all. Only an average of 2.9% of all segments actually applies the SACK opportunity, which was permitted by around 90% of all connections. It is interesting, that although 15.5% of the connection establishments advertise usage of the TS option, it just reappears in 9.3% of all segments. This might be caused by TCP servers not responding with the TS option set in their initial SYN/ACK. All other option kinds were observed with very low frequency.

3.3.2 TCP option values

Allman [9] reported about 90% of connections advertising an MSS of about 1460 bytes in the SYN segment, leaving 6% for larger MSS values, and another 5% for MSS values of about 500 bytes. An analysis of advertised values within the MSS option field in our data revealed that the major portion (93.7%) of the MSS values still lies between 1400-1460 bytes, thus close to the Ethernet maximum (1500-40 byte for IP and TCP headers). Values larger than 1460 bytes are carried by only 0.06% of the MSS options, with values up to the maximum of 65535. Values smaller than 536 bytes (the default IP datagram size minus 40) are carried by another tiny fraction (0.05%), including MSS values down

to zero. The 53,280 packets carrying small MSS values are send by 2931 different IP addresses. The major fraction of the <536 MSS values carries a value of 512 (87.5%), followed by 64 (2.4%) and 260 (1.3%). Values down to 265 bytes can be explained by standard active fingerprinting attacks, like nmap [15], whereas smaller values are more likely to be DoS exploits.

In Allman's data from 2000, Window Scale (WS) factors as high as 12 appeared, with zero as the main factor, accounting for 84%, followed by a factor of one with about 15%. In our contemporary data, WS factor values appear in the range of 0 to 14. The most common scale factor with 58% is zero, which should not be interpreted as real factor, but as an offer to scale while scaling the own receive window by 1. The major real scale factor appears to be 2, with 30.8% deployment. Other scale factors in recognizable fractions are 3, 1, and 6, applied in respectively 4.2%, 4.1% and 1.0% of all segments carrying a WS option. As a general observation, the WS option is applied much more effective now, most probably due to bandwidth increases and larger data transfers. A detailed look at diurnal behavior of WS option values revealed that traces at nighttime (2AM) carry constantly about 10% more scale factor values of 2, compensated by around 10% less factors of zero.

3.3.3 TCP option misbehavior

Connected to the analysis of TCP options, a couple of anomalies were encountered (Table 3(a)). The table shows only counts of packets, since the relative fractions are too small compared to the amount of total TCP segments. It should be mentioned that the differences between outgoing and incoming traffic lie typically in the order of a magnitude. Also diurnal differences can be observed, with non-working hours (2AM and 8PM) responsible for 67% of all reported anomalies.

Anomaly	2AM	10AM	2PM	8PM
Undef.Kind	1062	507	413	388
Invalid OL	1200	399	915	3020
Invalid HL	71	528	130	119

(a) TCP options and header lengths

Anomaly	2AM	10AM	2PM	8PM
RST+SYN+FIN	8	35	11	15
RST+SYN	25	70	43	27
SYN+FIN	4	22	8	9
Zero Flags	32	78	86	90
RST+FIN	10200	10988	14320	16334

(b) TCP flags

Table 3: Anomalies in TCP Headers

The first misbehavior experienced was the occurrence of undefined option types. Out of the 8bit range for TCP option kinds, only 26 are defined. From the remaining types almost all (228) have been observed. 522 distinct sources sent the 2370 undefined options observed, with 85% appearing on the incoming link. One single source sent 42% of these packets during the 20 minutes duration of one measurement at 2AM. Usage of a single destination port and 8200 different destination hosts within a one network prefix clearly indicate a scanning attack, even though only a minor fraction (6%) of the scanning traffic actually showed undefined options. The malformed packets carried instead of {MSS, NOP, NOP, SACK perm.} the option sequence of {MSS, random byte, random byte, 0, 0}. It seems likely that it is indeed the scanning software which is buggy and generates occasional malformed packets.

Another inconsistency encountered are option headers appearing to be valid while carrying option lengths that do not correspond to the total header length in the regular TCP header. 98.2% of the 5534 cases happened on the incoming link, with two sources responsible for 45% and 22% of such anomalous headers. The first source adds a SACK option with constant pattern to the TCP header, declaring an option header length of 180 bytes. This source was observed at 4 different days. The second source applies valid TCP options including an MSS value of 1460 during connection setup in SYN/ACK packets. However, also in the proceeding data packets an option of type 2 (MSS) appears, but this time followed by zeros, and thereby consequently advertising an option length of zero. According to the traffic pattern this source was a webserver. In total, 259 unique sources of this anomaly have been identified.

Finally, 848 TCP segments advertising header length values of less than 20 bytes were generated by 184 distinct sources, probably being DOS exploits. Again, the major fraction (91.3%) was observed in incoming traffic. 81.5% of the invalid values advertised a TCP header of zero length. The remaining 18.5% are evenly distributed between the remaining possible length values (in multiples of 4). The main source of zero byte TCP headers sends 351 such packets during a period of at at least 20 minutes. 351 unique destinations for 351 packets indicate a scanning campaign, this time to some well-known source port numbers (21, 23, 110, 80, 8080).

3.3.4 TCP Flags

Analyzing the TCP flag field, 10,972 ECN-setup SYN packets and just 800 ECN-setup SYN/ACK segments (RFC 3168) have been observed. The small numbers are consistent with earlier observations by Medina et al. [10], where only 0.2% of tested web clients advertise ECN capabilities. In section 3.2.1 we identified around 2.1 million ECN capable IP packets. This indicates that the few ECN enabled TCP connections represent large flows.

The urgent flag (URG) was set in only 663 segments. The acknowledgment flag (ACK) on the other hand was set in 98.6% of all segments, which is expected, since theoretically only the initial SYN packets should not carry an ACK flag. The push bit (PSH) was enabled in 22.4% of all segments.

In Table 3(b) we present packet counts for unexpected combinations of connection flags. The four first-listed anomalies have been seen in packets sent by 56 distinct sources. Such inconsistencies can easily be generated by port scanning tools like nmap [15]. We can rule out T/TCP as reason for SYN+FIN packets, since none of the 43 packets carried CC options (RFC 1644). The most frequent anomaly is connection termination with both, FIN and RST flags set. This was seen in 51,842 segments, send by 7576 unique source IP addresses. All connection flag anomalies are spread quite evenly over all measurements, with no particular sources to stand out.

4. SUMMARY AND CONCLUSIONS

In order to be able to present contemporary characteristics of Internet traffic, 148 traces of 20 minutes duration have been collected on a pair of backbone links in April 2006. The analysis revealed that IP options are virtually not applied and IP fragmentation is done to a minor extent (0.06%), with UDP accounting for most IP fragments. The latter observation stems from an increased employment of TCP Path MTU Discovery, which was shown to be even more dominating than reported earlier. Regarding packet size distribution, two findings should be noted. First, IP packet lengths of 628 bytes have become even more common than the default datagram size, with P2P traffic identified as likely source. Second, except for router traffic, jumbo packets are used for a single custom application only and are not seen otherwise. Even though these observations are limited to our measurements from a single point in the Internet, this summary about current behavior of network protocols helps to understand the influence of additional protocol features on Internet traffic and can contribute to an improvement of future simulation models.

Additionally, a number of anomalies and inconsistencies have been observed, serving as pointers to keep in mind for protocol and application developers. As one cause for the otherwise rare occurrence of IP fragmentation additional headers introduced by VPN have been identified, advising application developers to use smaller MSS values. Furthermore, one single long- duration UDP burst was observed while gathering protocol statistics. This was found to be an UDP DoS attack, undetected by the network management tools in operation. This indicates the need for continuous refinement of network monitoring policies. The magnitude of the burst also raises stability and fairness concerns, calling for addition of some kind of congestion control to UDP. Finally, several types of misbehaviors within IP and TCP headers have been discussed. The anomalies observed could be explained by three different causes:

- Buggy or misbehaving applications or protocol stacks
- Active OS fingerprinting [13]
- Network attacks exploiting protocol vulnerabilities

Even though all header anomalies observed are rare compared to the total number packets, their existence shows that developers need to carefully design implementations. Almost any possible inconsistency in protocol headers will appear eventually, thus network protocols and applications have to be designed and implemented as robust as possible, leaving no vulnerabilities.

Since access to traffic on highly aggregated links is still uncommon for researchers working on network security, our results form valuable input to related research on topics like large scale intrusion detection or traffic filtering. Besides quantifying the occurrence of different header anomalies 'in the wild', the results yield explanations for the origins of these commonly observed inconsistencies. Not every malformed packet header is necessarily part of attacking traffic, as proven by the example of the DNS server setting invalid fragmentation bits, but can also be introduced by improper protocol stacks. This information can be relevant when refining rule-sets for traffic filters, as applied in firewalls or network intrusion detection systems. Furthermore, knowledge about the nature of header anomalies can be interesting for researchers or developers creating stress tests for routers and other network components.

5. ACKNOWLEDGMENTS

The authors want to thank Pierre Kleberger for his kind technical support and Tomas Olovsson for his valuable and constructive comments throughout the MonNet project.

6. REFERENCES

[1] S. McCreary and K. Claffy, "Trends in wide area ip traffic patterns - a view from ames internet exchange," CAIDA, San Diego Supercomputer Center, Tech. Rep., 2000.

[2] N. Brownlee and K. Claffy, "Internet measurement," *IEEE Internet Computing*, vol. 08, no. 5, pp. 30–33, 2004.

[3] A. Householder, K. Houle, and C. Dougherty, "Computer attack trends challenge internet security," *Computer*, vol. 35, no. 4, pp. 5–7, 2002.

[4] S. Floyd and E. Kohler, "Internet research needs better models," ser. Comput. Commun. Rev. (USA), vol. 33. Princeton, NJ, USA: ACM, 2003, pp. 29–34.

[5] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area internet traffic patterns and characteristics," *IEEE Network*, vol. 11, no. 6, pp. 10–23, 1997.

[6] C. Shannon, D. Moore, and K. Claffy, "Beyond folklore: observations on fragmented traffic," *IEEE/ACM Transactions on Networking*, vol. 10, no. 6, pp. 709–20, 2002.

[7] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and C. Diot, "Packet-level traffic measurements from the sprint ip backbone," *Network, IEEE*, vol. 17, no. 6, pp. 6–16, 2003.

[8] K. Pentikousis and H. Badr, "Quantifying the deployment of tcp options - a comparative study," *IEEE Communications Letters*, vol. 8, no. 10, pp. 647–9, 2004.

[9] M. Allman, "A web server's view of the transport layer," *SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 5, 2000.

[10] A. Medina, M. Allman, and S. Floyd, "Measuring the evolution of transport protocols in the internet," *Computer Communication Review*, vol. 35, no. 2, pp. 37–51, 2005.

[11] A. Hussain, G. Bartlett, Y. Pryadkin, J. Heidemann, C. Papadopoulos, and J. Bannister, "Experiences with a continuous network tracing infrastructure," in *MineNet'05: ACM SIGCOMM workshop on Mining network data*. New York, NY, USA: ACM Press, 2005.

[12] J. Xu, J. Fan, M. Ammar, and S. Moon, "On the design and performance of prefix-preserving ip traffic trace anonymization," in *IMW '01: ACM SIGCOMM Workshop on Internet Measurement*. New York, NY, USA: ACM Press, 2001.

[13] T. Karagiannis, A. Broido, N. Brownlee, k claffy, and M. Faloutsos, "File-sharing in the internet: A characterization of p2p traffic in the backbone," UC Riverside, Tech. Rep., 2003.

[14] CiscoSystems, "Ipsec vpn wan design overview," Cisco Doc., 2006. [Online]. Available: http://www.cisco.com/univercd/cc/td/doc/solution/ipsecov.pdf

[15] Fyodor, "Nmap security scanner," 1998. [Online]. Available: http://insecure.org/nmap/index.html

PAPER II

Wolfgang John and Sven Tafvelin

Differences between in- and outbound Internet Backbone Traffic

TNC '07: TERENA Networking Conference
Copenhagen, DK, 2007

Differences between in- and outbound Internet Backbone Traffic

Wolfgang John and Sven Tafvelin
Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, Sweden
e-mail: {johnwolf,tafvelin}@chalmers.se

Abstract

Contemporary backbone-traffic is analyzed with respect to behaviour differences between inbound and outbound Internet traffic. For the analysis, 146 traffic traces of 20 minutes duration have been collected in April 2006, carrying 10.7 billion frames and 7.5 TB of data. Significant directional differences, among others found in IP fragmentation, TCP termination behaviour and TCP options usage, are pointed out and discussed on different protocol levels (IP, TCP and UDP). The analysis includes a focus on TCP connection properties, yielding P2P and malicious traffic as main reasons for the differences. The results are relevant for a better understanding of how applied network protocols are used in an operative environment. Furthermore, a quantification of malicious traffic provides related research fields, such as network security or intrusion detection, with important insights.

Keywords

Internet Measurement; Directional Traffic Differences; TCP Connection Analysis; Network Anomalies;

1. Introduction

The Internet, as emerging key component for commercial and personal communication, has in the recent years undergone a fast development and is still expanding. Unfortunately, this rapid development has left little time or resources to integrate measurement and analysis possibilities into Internet infrastructure, applications and protocols. However, the Internet community needs to understand the nature of Internet traffic in order to support research and further development [3]. One way to acquire better understanding is to measure real Internet traffic. In the MonNet project [10][24], the technical and legal complications of the measurement task were overcome resulting in packet-level traces of contemporary Internet traffic.

The MonNet traffic traces analyzed in this article have been taken from the OC192 backbone of the Swedish University Network (SUNET) during 20 days in April 2006. The links tapped provide not only a backbone for two major Universities, but also for a substantial number of student dormitories and research institutes. Additionally, the links carry exchange traffic with commercial providers due to a local exchange point inside Göteborg. Because of the high aggregation of the measured links, we believe that this recent data provides a valid footprint of Internet traffic characteristics in Sweden at the current time.

The chosen measurement point on the outermost part of a ring architecture makes the traces specifically suitable for highlighting directional differences. Put simply, the measurements were taken on links between the region of Göteborg and the rest of the Internet. This work therefore analyzes the contemporary data with respect to behaviour differences between in- and outbound backbone traffic. The presented traffic constitutes a medium level of aggregation, between campus-wide traffic and tier-1 backbone traffic. We believe that this type of network, with smaller local exchange points, represents an upcoming class of networks.

1.1. Related work

There are numerous articles about general Internet measurements [7][16][25], with only a few of them partly dealing with directional differences. Thompson [25] e.g. presented wide-area Internet traffic characteristics on nowadays rather outdated data in 1997. The data was recorded on a core-backbone and a transatlantic link, including figures about directional differences in packet sizes and byte volumes.

* This work was supported by SUNET, the Swedish University Network

In recent years, a few studies included discussions about directional differences, but typically only regarding specific properties. These articles are often based on unidirectional flow data and analyze a variety of datasets. The analyzed datasets are either collected at Tier-1 backbone level or on small campus or institute Internet gateways, so with either a low or very high level of aggregation. In his article about rapid model parameterization, Lan [14] showed differences between inbound and outbound traffic in terms of protocol mix and flow statistics, like flow size and duration. The data was recorded on the 100 Mbit/s Internet gateway of the USC Information Science Institute in 2001. Saroiu [22] analyzed different types of HTTP flows, recorded on two border routers of the University of Washington on 9 days in 2002. In this paper, WWW and P2P traffic carried over HTTP are contrasted, including a comparison of inbound and outbound flows. In his study about P2P properties in 2003, Gerber [8] was able to show that the IN/OUT traffic balance for P2P traffic on the border of a Tier-1 backbone is close to one. Kim [12] compared inbound and outbound flow statistics for different transport protocols, including flow, packet and byte ratios. The analysis was based on flow data collected in 2004 on the Internet routers of the POSTECH campus, a 2x100 Mbits/s Ethernet.

An interesting study based on packet-level traces was presented by Mellia [17]. Mellia analyzed traces collected on the Internet access link of the Politecnico di Torino campus LAN in 2000-2002. Besides presenting an automatic tool for statistical analysis of network traces, interesting results for IP and TCP characteristics are given, including a connection-level analysis of TCP.

1.2. Contribution of this work

Updated measurement results are crucial for a better understanding of how the applied technologies and protocols are used in an operative environment. In the present study, significant directional differences are pointed out and discussed on different protocol levels (IP, TCP and UDP). For TCP, the bi-direction packet level traces are reassembled to connections, in order to be able to conduct a detailed connection-level analysis. The presented results are destined for network engineers, network application developers and protocol designers in order to be able to optimize bandwidth efficiency and stability of future networks. The paper furthermore highlights network anomalies and inconsistencies, like attacking or scanning traffic. This is important knowledge, since improving the robustness of network applications and protocol implementations is gaining special importance. In fact, increasing bandwidth and growing numbers of Internet users have also lead to increased misuse and anomalous behaviour [9][13]. Knowledge of real-life traffic properties is also important for establishing more realistic simulation models [6]. Finally, some of the insights might as well bring up new research issues in related research fields, such as network security and intrusion detection. The contributions of this work are relevant, because:

- the analysis is based on updated, contemporary data
- the data was collected on links representing medium traffic aggregation, a class of networks not previously studied in the same extent
- packet-level traces allow a more detailed analysis than sampled flow-level data (e.g. TCP options)
- the presented bi-directional TCP connection analysis reflects real connections more closely than traditional flow level analysis
- the results provide a complete view of directional differences on different levels (IP, TCP, UDP)
- the special focus on network anomalies is especially important in the light of increasing amounts of network attacks

The paper is outlined as follows. Section 2 describes the methodology of collecting, pre-processing and analyzing the traces. Then some general traffic properties are presented in section 3. Next, sections 4, 5 and 6 quantify directional differences observed on different protocols levels (IP, TCP and UDP). Finally, in section 7, different traffic anomalies and inconsistencies found on the protocol levels are summarized, followed by concluding remarks about the main findings.

2. Methodology

2.1. Collection of traces

We collected our traces on the outermost part of an SDH ring running Packet over SONET (PoS). The traffic passing the ring to (outbound) and from (inbound) the main Internet is primarily routed via our tapped link, as confirmed by SNMP statistics. Simplified, we regard the measurements to be taken on links between the region of Göteborg, including exchange traffic with the regional access point, and the rest of the Internet as discussed earlier in section 1.

We use optical splitters on two OC-192 links, one for each direction. The splitters are attached to two Endace DAG6.2SE cards sitting in identical Dual-Opteron servers. The servers use a 6 disk SCSI Raid0 to keep up with the speed of the 10 Gbit/s links. The DAG cards are configured to capture the first 120 bytes of each frame to ensure that the entire network- and transport header information is preserved. The two DAG cards are chained together with help of the DAG Universal Clock Kit (DUCK), with one card serving as synchronisation input for the second card, resulting in time synchronisation typically between ± 30ns [5].

The collection of the data was performed between the 7th of April, 10:00 and the 26th of April 2006, 10:20. During this period, we simultaneously for both directions collected four traces of 20 minutes each day at identical times. The times (02:00, 10:00, 14:00 and 20:00) were chosen to cover business, non-business as well as night time hours. Due to measurement errors in one direction at four occasions we have excluded these traces and the corresponding traces in the opposite direction.

2.2. Processing and analysis

After storing the data on disk, the payload beyond transport layer was removed and the traces were sanitized and desensitized. This was mainly done by using available tools like Endace's dagtools and CAIDA's CoralReef, accompanied by own tools for additional consistency checks, which have been applied after each pre-processing step to ensure sanity of the traces. 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. During our capturing sessions, the DAG cards discarded a total of 20 frames within 12 different traces due to receiver errors, which includes HDLC CRC errors. Surprisingly, another 71 frames within 30 different traces had to be discarded after the sanitization process due to IP checksum errors.

By desensitization we mean the removing of all sensitive information to ensure privacy and confidentiality. The payload of the packets was removed earlier, so we finally anonymized IP addresses using the prefix preserving CryptoPAN [27]. After desensitization, the traces were moved to a central storage server. First, an analysis program was run on each trace to extract cumulated statistical data. As a second step, per-connection TCP analysis was conducted on merged, then bidirectional traces. More details on the connection analysis are described in beginning of section 5.

3. General traffic characteristics

As summarized in table 1, the 146 analyzed traces sum up to 10.68 billion PoS frames, containing a total of 7.53 TB of data. In his study on campus wide traffic, Kim [12] reported about a 1:1 ratio between outbound and inbound traffic for packets numbers, but an 1:1.38 inequality for traffic volume due to the “net provider” status of University networks. In the our data, no significant difference between neither, packet counts nor volumes, can be observed. This even distribution of traffic proves the higher level of aggregation and underlines the relevance of the presented data, representing Internet backbone traffic.

The frames contain in 99.97% of the cases IPv4 packets, which sum up to 99.99% of the carried data. The remaining traffic consists constantly of around 1200 IPv6 BGP Multicast messages, 8 CLNP routing updates (IS-IS) and 1 Cisco Discovery Protocol (CDP) message per minute. The results in the remainder of this paper are based on the IPv4 traffic only.

	Packets		Data	
	Count	%	Volume	%
Total	10.68E+9	100.00%	7.53 TB	100.00%
Outbound		48.74%		49.16%
Inbound		51.26%		50.84%

Table 1: Traffic amount of data captured

	Total		Outbound		Inbound	
	Inside	Outside	Source	Dest.	Dest.	Source
Total	634E+3	22.0E+6	275E+3	19.2E+6	490E+3	19.8E+6
TCP	408E+3	05.0E+6	176E+3	04.3E+6	310E+3	04.5E+6
UDP	484E+3	19.2E+6	175E+3	16.4E+6	384E+3	16.9E+6
Rest	155E+3	01.9E+6	024E+3	01.1E+6	146E+3	01.0E+6

Table 2: Distinct IP addresses seen

4. IP level

In this section out- and inbound traffic on the network layer level of the Internet Protocol (IP) is compared. This comparison includes the transport protocol mix, IP packet size distribution and IP fragmentation.

To start with, table 2 gives some scale to the aggregation level of the links. The numbers of distinct IP hosts seen within (inside) and outside the region of Göteborg are summarized, where outbound sources and inbound destinations are regarded as inside, and the opposite way around as outside. Note that the sum of the numbers exceeds the total numbers, since one host can obviously be both source and destination for packets of several transport protocols. As expected, the amount of hosts inside the region is outnumbered by hosts seen outside “in the Internet”. Nevertheless, there is a surprisingly high number of hosts inside, considering that the numbers of hosts at the three main customers of the links (2 major universities and the regional network for student dormitories) do not exceed 7,000 each. Indeed, these main customers sum up to about 21,000 sources of outbound TCP connections. The remaining 150,000 outbound sources belong to different providers connected to the exchange point. The amount of inbound destinations is much larger due to incoming scanning traffic. As an example, the 16 bit address ranges of the two Universities are scanned in their entirety (2x65,534). The vast amount of UDP hosts outside was found to be due to short UDP sessions caused by P2P overlay networks, which will be discussed in more detailed in section 6.

It has to be noted that even though the hosts of the three main customers represent a minor part (13%) of the observed IP addresses inside the region of Göteborg, a majority of the traffic (around 85%) consists of packets to or from these hosts.

4.1. Transport protocol breakdown

The protocol breakdown, summarized in table 3, once more confirms the dominance of TCP traffic. Compared to earlier measurements [7][16][23][25], the fractions of both TCP data volume and packet counts have even increased slightly. In the table, fractions of packets and data carried in the respective protocol are in % of total IPv4 traffic for the corresponding direction. Ratios between out- and inbound traffic are shown in parentheses, summing up to 100 for each protocol.

TCP packets and data show an equal ratio, as it was the case for the total traffic. In Kim’s report [12], outbound traffic carried 1.44 times more data than inbound traffic. We believe that this behaviour is not observed in our data since the traffic of diverse network types aggregating on the links measured cancel out the typical client-server imbalance (small requests, large data replies). UDP data on the other hand shows almost the same ratio (38:62) in favour of inbound data volumes in our data as previously reported by Kim. This is caused by multimedia traffic (mainly RTP) over UDP, which is more common to be served on hosts on the Internet. An interesting observation can be made for UDP packets, with an unexpected large amount of outgoing packets. A closer look reveals that three consecutive measurements carried up to 58% UDP packets, as shown in table 4. This indicates a potential single UDP burst of 14-24 hours of time. A detailed analysis shows that the packet length for the UDP packets causing the burst was just 29 bytes, leaving a single byte for UDP payload data. These packets were transmitted between a single sender and receiver address with varying port numbers. After reporting this network anomaly, the network support group of a University could identify the culprit host. This was a web server that had been exploited through a known vulnerability in a PHP script. Consequently, a UDP DoS script was installed and could run undetected, since

the network management tool was monitoring amount of per-flow data only, but not the number of packets. Although TCP data was still predominant, we believe that a dominance of UDP packets over such a time span could potentially lead to TCP starvation and raise serious concerns about Internet stability and fairness. When removing the three traces with this outstanding network event from our data, UDP packets showed the same ratio as the TCP and the overall data. Due to the small packet sizes, the ration of UDP data kept almost unchanged (36:64).

ESP traffic seems to experience a typical client-server pattern with even packet ratio, but uneven data proportions. The hosts mainly responsible for this type of traffic will be discussed again in section 4.3. An explanation for the dominance of outbound traffic for ICMP could be the large amount of incoming network attacks as shown later, triggering ICMP responses from routers and firewalls.

	Fraction of Packets in %				Fraction of Data in %			
	outbound		inbound		outbound		inbound	
TCP	90.62	(48.1)	93.14	(51.9)	97.76	(49.5)	96.57	(50.5)
UDP	8.87	(56.8)	6.40	(43.2)	2.03	(37.8)	3.23	(62.2)
ESP	0.23	(52.5)	0.20	(47.5)	0.12	(66.5)	0.06	(33.5)
ICMP	0.22	(61.9)	0.13	(38.1)	0.02	(60.7)	0.02	(39.3)
GRE	0.05	(51.8)	0.05	(48.2)	0.07	(73.0)	0.02	(27.0)

Table 3: Protocol mix (ratios per protocol in parenthesis)

		outbound	
Date	Time	Packets	Data
2006-04-16	14:00	6.8%	1.7%
2006-04-16	20:00	40.6%	5.1%
2006-04-17	02:00	51.9%	6.1%
2006-04-17	10:00	58.1%	7.1%
2006-04-17	14:00	5.7%	1.8%

Table 4: UDP burst

Packet size	total	outbound	inbound
20-39	1.50%	2.96%	0.11%
40-60	38.72%	37.26%	40.12%
576	0.96%	0.60%	1.29%
628	1.76%	2.05%	1.49%
1300	1.11%	1.20%	1.01%
1400-1500	38.01%	37.66%	38.34%

Packet size	total	outbound	inbound
20-39	0.14%	0.18%	0.11%
40-60	39.25%	38.41%	40.02%
576	0.98%	0.63%	1.30%
628	1.79%	2.12%	1.49%
1300	1.13%	1.25%	1.01%
1400-1500	38.53%	38.62%	38.45%

Table 5: Major modes of IPv4 packet size distribution for all data (left) and without UDP burst (right)

4.2. Packet size distribution

While cumulative distribution of IPv4 packet sizes was reported to be trimodal in earlier measurements [7][16][23][25], more recent studies showed that it has changed to be rather bimodal [21]. The two major modes are small packet sizes just above 40 bytes (TCP acknowledgements) and large packets around 1500 (Ethernet MTU). The previous third mode of 576 bytes (default size according to RFC 879) has in our data decreased to less than 1%. Furthermore, we found that two other notable modes appeared at 628 bytes and 1300 bytes. In table 5 the major modes are summarized, with an extra table excluding the above mentioned UDP burst. As discussed in a prior study on the SUNET datasets [10], the mode at 628 bytes is an artefact of 'TCP layer fragmentation' applied by file sharing protocols like Bittorrent or DirectConnect, where 628 byte large packets typically appear after full sized packets in order to add up to 2KB blocks of data. The mode at 1300 bytes could be explained by the recommended IP MTU for IPsec VPN tunnels [4].

The studies of Thompson, Kim and Mellia [12][17][25] report about directional differences in packets sizes on two different levels of link aggregation, both caused by the classical client-server pattern. In contrast, in the SUNET data the two main modes for small and large packets show no significant directional differences. This might be due to two different reasons:

- since Thompson's report of 1997, network applications have undergone some fundamental developments
- compared to the campus-wide data of Kim and Mellia, our backbone data contains a higher aggregated traffic mix

Directional differences however can be observed for two other packet sizes. The differences between fractions of 628 byte sized packets are likely to be caused by popular P2P servers inside Göteborg's student network. It is well known that DirectConnect, but also Bittorrent are especially popular in Sweden, and

consequently also in the region of Göteborg. The cause for the difference in the default datagram size of 576 bytes is not obvious, but we think it might be caused by a better utilization of the Path MTU discovery feature in the comparable well configured hosts inside University and student networks.

4.3. IP fragmentation

Earlier studies of McCreary and Shannon [16][23] indicated an increase in the fraction of IP packets carrying fragmented traffic of to up to 0.67%. We found a much smaller fraction of only 0.065% of fragmented traffic in the analyzed data, as shown in table 6. It can be noted that 72% of the fragmented traffic in our data is transmitted during office hours, at 10AM and 2PM. While Shannon, analyzing data of three different locations in 2001, found that fragmented data was equally distributed between out-and inbound data, the amount of fragmented traffic on the SUNET inbound link is about 9 times higher than on the outbound one. Where UDP and TCP is responsible for 97% and 3% respectively of all incoming fragmented segments, they just represent 19% and 18% of the outgoing. The remaining 63% outgoing fragmented traffic turned out to be IPsec ESP traffic (RFC 4303) between exactly one source and one receiver at working hours on weekdays. Each fragment series in this connection consists of one full length Ethernet MTU and one additional 72 bytes fragment. This could easily be explained by an unsuitably configured host/VPN combination transmitting 1532 byte (1572-40 byte additional IP and TCP header) instead of the Ethernet MTU due to the additional ESP header. The dominance of UDP among fragmented traffic is not surprising since Path MTU Discovery is a TCP feature only.

	Total	outbound	inbound
Total	0.065% (100.0%)	0.014% (100.0%)	0.113% (100.0%)
TCP	(4.5%)	(18.0%)	(2.9%)
UDP	(88.6%)	(18.8%)	(97.1%)
ESP	(6.8%)	(63.1%)	(0.0%)

Table 6: Fractions of IPv4 fragments

The first approach to explain the differences is based on the fact that the probability for a packet to be fragmented is increasing with each hop. According to a TTL analysis of the fragmented traffic, the average hop count for outbound traffic was 6.77, whereas the average hop count for inbound traffic was 9.43. This alone does not seem to be significant enough to explain the imbalance between inbound and outbound fragments. We believe that another possible explanation could again be the fact that SUNET and its connection networks are very well configured and administered compared to Internet standards.

5. TCP level

In order to conduct a detailed connection level analysis on TCP, we merged the tightly synchronized unidirectional traces. From the resulting bidirectional traces an analysis program collected per-connection data, including 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. We define a connection by the classical tuple of IP addresses and ports for source and destination. A TCP connection starts with the observation of the first SYN segment and is closed by either one FIN segment for each direction or one RST segment. Additional SYN segments for one tuple can sometimes be seen in the same direction, most commonly within scanning campaigns. In this case, further “connections” are opened within the analysis program in order to record the pure SYN packets separately. The following non-pure-SYN packets are always recorded within the most recently opened connection. We decided not to use a timeout threshold for unclosed connections, since our traces are limited to 20 min duration anyhow.

A significant part of the traffic is routed asymmetrically, due to hot-potato routing. 8% of the TCP data was sent via the outgoing link, without any corresponding TCP packets seen on the incoming. Asymmetrical

traffic on the incoming link was even more common, accounting for 20% of the observed TCP data. Knowing the prefixes of the SUNET network segments in the area of Göteborg, it was possible to show that around 14% of the TCP data is actual transit traffic with neither source nor destination being SUNET customers inside Göteborg, entering the links via the local exchange point. Of the transit traffic, 67% was asymmetrical traffic, which means that 1/3 of all asymmetrically routed traffic is transit traffic as well.
In the following subsections, first, TCP connections are classified according to their connection setup and termination behaviour. Then, connection properties like packet count, byte size and lifetime are analyzed with respect to connection direction. Finally, TCP options are discussed in the rather novel approach of per-connection information for SYN requests and replies.

5.1. Connection breakdown

The following tables summarize the connection breakdown for TCP in all 146 traces. The analysis database recorded a total of 72.6 Million connections according to our definition. Additional 8.9 million bidirectional flows do not include an initial SYN segment, which means that they either start before the measurement times or have asymmetrical properties. One million of these flows include no SYN, FIN or RST segments but show packets in both directions, which means that about 3.4% of the established connections last longer than 20 minutes. However, this small number of long lasting connections carries about 34% of the total TCP data. This is not unexpected, given the observations of Brownlee [2], saying that flows longer than 15 minutes carry more than 50% of the traffic on a link. According to their destination port numbers, the long lasting connections typically carry traffic of different P2P protocols and popular messenger services.
The following analysis is performed on TCP connections with initial SYN segments.

	total		outbound		inbound	
	Count	%	Count	%	Count	%
TCP connections	72.6E+6	100.00%	28.0E+6	38.56% (100.00%)	44.6E+6	61.44% (100.00%)
rejected	44.3E+6	60.99%	12.3E+6	(44.04%)	32.0E+6	(71.63%)
established	28.3E+6	39.01%	15.7E+6	(55.96%)	12.7E+6	(28.37%)

Table 7: TCP connection attempt breakdown

rejected connections	44.3E+6	100.00%	12.3E+6	27.84% (100.00%)	32.0E+6	72.16% (100.00%)
scanning - no reply	34.8E+6	78.66%	08.2E+6	(66.74%)	26.6E+6	(83.26%)
asymetric traffic	04.8E+6	10.84%	02.2E+6	(17.94%)	02.6E+6	(8.10%)
scanning - RST reply	04.3E+6	9.81%	01.7E+6	(13.83%)	02.6E+6	(8.25%)

Table 8: Rejected connection breakdown (no 3-way handshake)

Table 7 presents the total of all TCP connections with initial SYN segments. We define established and rejected connections as connections experiencing a proper 3-way handshake or not, respectively. Outbound in this context means that the initial SYN packet was sent on the outbound link. Inbound consequently means that the connection establishment was initiated outside the region of Göteborg. The tables 8 and 9 summarize the termination properties for rejected and established connections. In the tables, the first line represents the vertically summed values for each respective column of absolute packet counts or relative fractions. The fractions of out- and inbound connections in relation to the total amount of connections are additionally given in the first line, summing up to 100% horizontally.
Arlitt [1] quantified different TCP connection states based on the campus wide traffic recorded at the University of Calgary between 2003 and 2004. He quantified rejected connections with about 25-30% of all TCP connections. Our contemporary data includes much more unsuccessful connection attempts, as shown in table 7. A major difference between the numbers of rejected outbound and inbound initiated connections is evident in table 8. The large amount of unreplied SYN packets on the incoming link was already indicated earlier, when discussing the numbers of distinct IP addresses appearing on the incoming link. These are mainly attacks trying to exploit well known vulnerabilities on ports commonly used by Trojans. The scans

often cover the entire IP ranges of the connected networks inside Göteborg and are likely to be destined for non existing endpoints. Entrance routers to the specific network typically drop this kind of packets, which explains the absence of response packets. In some cases an ICMP response might be triggered, which would explain the larger number of outgoing ICMP packets according to table 3. Regardless of the much higher number of incoming scans, there is also a substantial number of outgoing unreplied connection attempts. More than 70% of the 8.2 Million attempts are sent by hosts within the student network. Note that not all of these attempts are necessary network scans. There is a large fraction of non-malicious outbound connection attempts to non existing hosts, resulting in unsuccessful connection attempts. This is often observed for P2P traffic, where unreliable file-sharing peers are common.

In cases where scanning attempts reach existing hosts on arbitrary port numbers, host-based firewalls should preferable drop the packets, but might in some cases reply immediately with an RST packet. This behaviour is more than twice as common for hosts in the student network as compared to hosts in University networks, which indicate that private Internet hosts are less carefully configured.

Asymmetric traffic was included in the summary for rejected connections (table 8) for reasons of completeness. Naturally, asymmetric traffic can not experience a bidirectional 3-way handshake, which means that we cannot consider this traffic as being established.

	total		outbound		inbound	
	Count	%	Count	%	Count	%
established connections	28.3E+6	100.00%	15.7E+6	55.21% (100.00%)	12.7E+6	44.68% (100.00%)
proper closing (2xFIN)	19.0E+6	66.99%	11.4E+6	(72.87%)	07.6E+6	(59.71%)
FIN and RST outbound	03.2E+6	11.21%	542E+3	(3.46%)	02.6E+6	(20.81%)
FIN and RST inbound	01.7E+6	6.06%	711E+3	(4.54%)	01.0E+6	(7.93%)
single RST	02.2E+6	7.71%	01.6E+6	(9.98%)	620E+3	(4.89%)
FIN, RST in counter dir.	01.2E+6	4.11%	889E+3	(5.67%)	276E+3	(2.18%)
unclosed	01.0E+6	3.63%	487E+3	(3.11%)	540E+3	(4.27%)

Table 9: Established connection termination breakdown

In table 9 finally the 28.3 Million connections with proper 3-way handshake observed are split up into different termination behaviours per direction. Considering the quite even distribution of TCP traffic volumes (table 1) it is somewhat surprising to see around 10% more outbound than inbound established connections. These differences in connection counts are cancelled out in the high level summary by differences in connection properties, as presented in the next subsection.

A major fraction (67%) of the established connections is closed properly by FIN segments in each direction, which seems to be quite low, considering that TCP resets should be a rare event according to the TCP standard (RFC 793). On the other hand, a prior study by Arlitt [1] highlighted that TCP connections are becoming more likely to be closed by RST segments (15%), mainly due to irregular web server and browser implementations. Comparing the behaviour of in- and outbound connection in our data we find that connections opened from inside Göteborg are more likely to be closed by proper FIN handshakes. This is compensated by a higher number of connections involving RST segments on the incoming link. While single RSTs in either direction can still be regarded as proper connection termination, the number of connections closed by FIN, followed by additional RST segments is surprisingly high (more than 30% on the inbound connections), even when considering Arlitts results. In fact, the fractions of connections closed by both FIN and RST segments sent by the client (the originator) are close to Arlitts numbers. (3.5% and 7.9% resp.) and are indeed mainly caused by web traffic. The main surprise is the large numbers of connections terminated by FINs and RSTs sent by the server (the responder), which are unproportionally large for inbound connections, meaning that they are closed by servers inside Göteborg. As main source of this behaviour a handful of hosts inside the student network could be identified, according to their port

numbers serving different kinds of popular P2P protocols. This reset behaviour is probably used to reduce the CPU and memory overhead introduced by connections entering the TIME_WAIT state on peers [1].
The 3.6% of unclosed connections lies close to the fraction of long-lasting connections, quantified in section 5.1. These unclosed connections are indeed mainly long lasting flows, and consequently carry almost 50% of all data carried by established connections. While 50% of these unclosed, long lasting incoming connections show destination port numbers of popular P2P protocols, the same port numbers account only for 10% of the outbound connections.

In addition to the high number of connections consisting of one SYN segment only, we also observed as many as 57 Million connections consisting of RST segments only (not shown in the tables). Of these single RST segments, 96% are seen on the outbound link, almost entirely in asymmetrical fashion, without any incoming segment triggering the resets. Only a handful source/destination pairs are responsible for these segments during short periods of time, so the first suspicion was that this could be reset attacks [26]. However, closer investigation showed no variations in sequence numbers or no other typical symptoms, so TCP reset attacks can be ruled out. We believe that the outbound link could be the return path for an asymmetrical routed denial of service (DoS) attack, generating the observed RST segments. Still, it is surprising that no similar behaviour could be observed to the same extent on the symmetrical routed data.

5.2. Quantification of P2P traffic

Since we expect P2P to have a huge impact on traffic characteristics, we tried to quantify P2P traffic for each direction with a simple port number analysis. Even though it is well known that P2P traffic is trying to hide itself and that port number methods strongly underestimate actual numbers [11][18], we believe that this analysis is still valid for comparing amounts of P2P connections between directions.
A list of common port-numbers for popular file-sharing protocols was identified, specifically for different DirectConnect, Bittorrent, Edonkey and Gnutella implementations. According to these port-numbers, outbound P2P connections carry around 13% of P2P packets and data, while for inbound connections this fraction is about twice as large with around 25%. Note, that these large volumes of data are carried by a small number of connections (less than 1%). Beside the probably quite large underestimation of these numbers, they indicate that P2P traffic is in fact at least about 2 times more common among inbound connections. The high amount of inbound established P2P connections, as already indicated in section 5.1, could be the result of a number of popular P2P peers inside Göteborg. Another possible explanation could be an increasing use of modern P2P clients (like RevConnect) inside Göteborg, triggering reverse connections from peers outside, on the Internet.

5.3. Connection properties

This section provides detailed information about different connection properties such as lifetime, size and packet count. The analysis deals only with bidirectional connections which have been established by a 3-way handshake. Ordering the TCP connections by data volume and number of packets carried shows that a small number of top connections accounts for most of the data and packets. This indicates the characteristically 'elephant and mice phenomenon', saying that the majority of Internet data is carried by a small percentage of large flows, so called elephants [15][20]. More specifically, outgoing connections appear to have less pronounced elephants, since it needs 0.08% and 0.17% to carry 50% of the total amount of data and packets respectively for outgoing connections, while only 0.07% and 0.14% are sufficient for 50% for inbound connections. This directional difference can be described even more clearly, considering that 3.9% of the outbound and as few as 0.9% of the inbound connections carry 90% of the data, and 26.3% and 12.2% respectively carry 90% of the packets seen in the particular direction.
Generally, artefacts of the client-server pattern (small requests, large data replies) can be observed for connections established in both directions. While outbound connections yield an average ratio of 1:1.6 in

favour for incoming data, inbound connections show a higher ratio of 1:1.86 in favour of outgoing data. This means that the smaller number of inbound connections (around 45% of all connections) carries more data and more packets primarily in outgoing direction, according to the client-server pattern. This imbalance is cancelled out to an almost even ratio in the high-level view of sections 3 and 4. The imbalance in connections properties is mainly caused by the larger fraction of heavy incoming P2P connections.

The differences between in- and outbound connections are summarized in table 10 by means of statistical properties. In the table, mean, standard deviation (σ), median and 80th percentile (P80) are given for different connections properties per direction of the initial connection establishment. While mean and σ of connection lifetimes appear to be quite similar for both directions, the values for sizes and packet counts are significantly larger for inbound connections. It needs to be noted that some of the values and figures in this subsection are somewhat biased since the traces are limited to 20 min of duration. Long-lasting connections, which are likely to be elephants, are therefore not taken into account to the full extent. Especially the values for mean and σ can therefore to be considered as an underestimate, while median and P80 are less biased.

In order to be able to better interpret median and 80th percentile, we included figures for the distributions of connection lifetimes, sizes and packet counts. Figure 1 illustrates distribution of lifetimes in bins of 1sec. The magnified figure presents the first 25 seconds, with higher resolution of 15.6 ms bins. Figure 2 shows connection size distribution, summarized in bins of 1Kbyte. The insert magnifies the first 9 Kbytes with 20 Byte bin-size. Figure 3 finally illustrates packets counts, including magnification for the first 100 packets.

Property		mean	σ	median	P80
Lifetime in sec	out	18.2	60.7	1.8	16.6
	in	17.3	65.8	0.6	24.8
Size in Kbytes	out	61.0	2362	1.1	2.9
	in	81.5	3298	1.9	8.9
Packet Count	out	81.5	2289	11.5	22.0
	in	113.0	3538	11.5	21.0

Table 10: Statistical properties of TCP Conn.

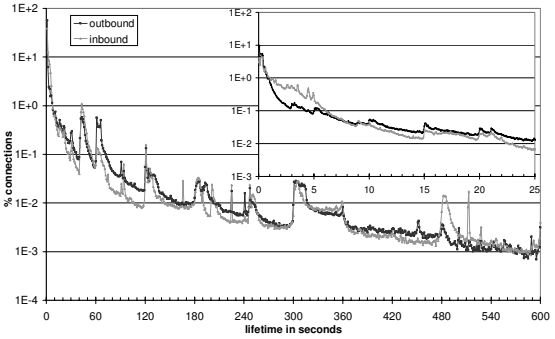


Figure 1: Conn. lifetimes with 1sec bins (15.6ms bins)

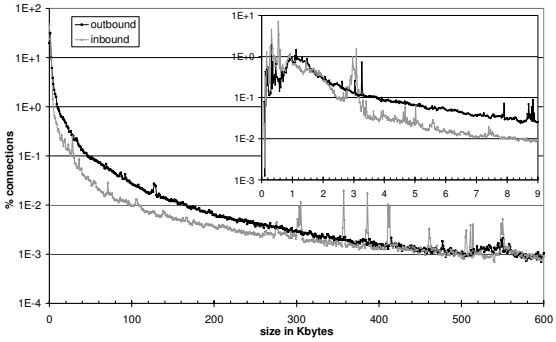


Figure 2: Conn. sizes with 1Kbyte bins (20 Byte bins)

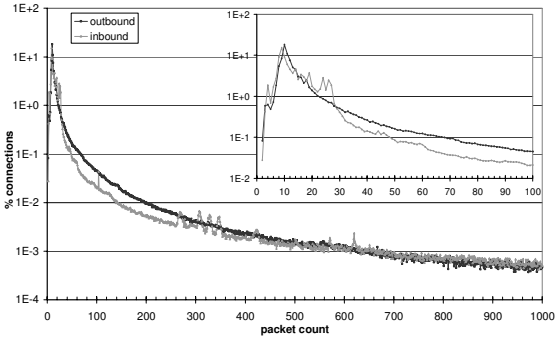


Figure 3: Packets per connection

Mori [19] presented mean values for flow durations on web and P2P flows extracted from inbound campus traffic in 2002. Web traffic yielded 9.5 sec mean, while P2P flow result in a mean of 307 sec. Projecting the values to our data, it can be concluded that the mean values of around 18 sec are a hybrid between web and P2P traffic, which is in fact the case due to University traffic on one hand and private student traffic on the other hand. Considering the underestimated nature of our values, it again indicates a quite substantial amount of long lasting P2P traffic on the measured links. Other studies, including Kim, Lan and Zhang

[12][15][28], presented cumulative distribution figures, reporting of median values of about 1sec and P80 values of around 10 sec. In the SUNET data especially the 80th percentile is significantly larger, again proofing that connections in contemporary traces tend to be significantly longer due to an increased amount of P2P traffic. This property is more pronounced for inbound connections when comparing the P80 values for connection lifetime. Surprisingly, inbound connections do not only tend to be longer, but are also more likely to be shorter than 5 seconds compared to outbound connections. This is indicated by the median values, but can be seen nicely in the magnification of figure 1. The large number of incoming connections in this region can be explained by rejected login attempts on application level, like SSH or SMTP. In general, figure 1 shows a number of protocol timeouts, typically close to half minute or minute borders. For most of the times, the fractions of inbound connections lie below the outbound ones, which is compensated by a higher number of long lasting flows, as discussed earlier.

Regarding connection sizes, Mori [19] also presented mean values with 20.6 Kbytes for web flows, and as large as 5.8 Mbytes for P2P flows. As for lifetimes, the mean values for the presented data lie in between these extreme values. Earlier studies reported about median values of around 1 Kbytes and P80 values of between 1 and 10 Kbytes [15][28], which is similar to our findings. Even though in contrast to connection lifetimes, both median and P80 value are larger for inbound connections, there are peaks in the magnification of figure 2 for incoming connection sizes below 1Kbyte and around 3 Kbytes. According to a port analysis, the former stems from connections trying to exploit a known security hole in some MS SQL server versions on a handful of hosts inside Göteborg, while the latter can be explained by unsuccessful SSH login attempts, probably mainly intrusion attempts as well. Generally, figure 2 shows that inbound connections tend to be less likely to carry small amounts of data, which again indicates that there is a higher number of “elephants” carrying a lot of data on the incoming link. This seems to be connected to the similar behaviour found for connection lifetimes, even though there is not necessarily a strong correlation between duration and size, as reported by Lan and Zhang [15][28]. The spikes for inbound traffic seen in figure 2 between 300 and 550 Kbytes are results of connections from a single host to one host on destination port 2135. This is rather a special application than another security exploit, since except these small connections there are also a larger number of connections carrying a large amount of data between these hosts.

As illustrated in the magnification of figure 3, connections with less than 20 packets show very similar patterns for both directions, consequently resulting in similar median and P80 values. Nevertheless, the differences in the mean values as well as the lower values for the inbound graph in figure 3 shows that packet counts are to some degree correlated with connection sizes. As for connection sizes, the spikes between 20 and 30 packets are artefacts from unsuccessful SSH logins, and the spikes between 300 and 360 stem from the unidentified connections to port 2135.

5.4. TCP option usage

In earlier work, TCP options analysis was typically done by counting occurrences of different TCP options in all SYN and SYN/ACK segments seen in packet-level traces [10][21]. In our current work, the thorough connection analysis allows us to give better insight into options advertisements between clients and servers within single TCP connections. Since this analysis is focused on proper established connections only, attacking and scanning traffic, which might bias simple counts of SYN segments, are filtered out.
Table 11 summarizes TCP option employment for the four major TCP options types typically advertised during connection establishment. Fractions of connections carrying the particular option in SYN or SYN/ACK segments are given, split up for outbound and inbound established connections. The third column presents the fractions of connections advertising the option in both initial segments, hence actually establishing the connection with the specific optional feature.

	MSS			WS			SACK			TS		
	SYN	SYN/ACK	both	SYN	SYN/ACK	both	SYN	SYN/ACK	both	SYN	SYN/ACK	both
outbound	100.00%	99.59%	99.59%	19.36%	15.46%	15.46%	93.67%	69.70%	69.70%	16.50%	12.32%	12.32%
inbound	99.94%	99.92%	99.85%	24.33%	23.85%	23.83%	97.22%	90.40%	90.38%	19.72%	18.51%	18.50%

Table 11: TCP options for inbound and outbound connections

In general, the numbers are in range of the reported values of the previous studies. The maximum segment size option (MSS) is used extensively by clients and servers for both directions. To our surprise, the window scale (WS), timestamp (TS) and selective acknowledgement permitted (SACK) options on the other hand are about 1.5 times more common among inbound connections. Looking at the destination port numbers for these connections, the difference can be explained by a much more diverse mix of applications among inbound connections in favour of primarily web traffic on port 80 in outgoing connections. The incoming connections include large fractions of recognized P2P protocols, but also substantial amounts of SMTP, SSH and MS SQL sessions, which are mainly break in attempts as discussed in section 5.3. These protocols are often used to carry more data than conventional web traffic, so it seems natural that clients and servers are interested in optimizing throughput by use of these TCP options.

6. UDP level

Since UDP offers no connection establishment or termination, we defined UDP flows as the sum of bidirectional packets observed between a specific tuple of source and destination IP and port numbers, taking advantage of the timeout value of 20 min given by the trace duration. In the 2x73 network traces, 68 million such UDP flows have been observed, carrying around 7% of the packets and only 2-3% of the data
Interestingly, 51 out of the 68 Million UDP flows (76%) carry less than 3 packets in either direction. Our first guess, that classical UDP services like DNS and NTP would be primarily responsible for these flows, proved to be wrong. In fact, only 5% and 1.7% of the small UDP flows serve DNS or NTP requests, respectively. On the other hand P2P overlay networks, such as Kademlia or other distributed hash table (DHT) protocols, are responsible for at least 18% of these small flows, where we expect this naïve port analysis to be a huge underestimate again. The purpose of these overlay networks is to keep the peers routing tables updated in a completely decentralized fashion. This is done periodically by sending DHT "pings" in small UDP packets, replied by the recipient. No significant difference between inbound and outbound DHT queries could be observed, which makes sense when considering the type and the nature of these overlay networks.
Based on the simple port classification, different common network attacks on UDP port numbers for MS SQL, MS messenger "spam" or Netbios were found to be responsible for at least in 8% of the 51 Million short flows. These flows consisted in more than 90% of the cases of one inbound packet only, sometimes performing scans on entire IP ranges.
The two main sources for UDP flows, P2P overlay networks and attacking traffic, finally also explain the extreme amount of distinct IP addresses seen on the outside of the links measured (presented in table 2) since P2P network span the entire globe and experience a very high fluctuation in peering partners.

7. Summary and Conclusions

We presented directional differences found on recent packet level traces taken on links with medium aggregation level, carrying traffic from two major Universities, about a dozen of large student dormitories and a local exchange point. Since access to contemporary traffic on highly aggregated links is still uncommon, we believe that this study can contribute to a better understanding of the changing behaviour of the Internet. After short discussions about the two main factors responsible for the observed directional differences in our traces, malicious traffic and P2P traffic, this paper will be closed with summarizing conclusions.

7.1. Malicious traffic

Already the protocol breakdown revealed one outstanding long-duration UDP DoS attack originated within a major University in Göteborg, due to an DoS script injected from outside by exploitation of a known vulnerability. The fact that this attack was undetected by the network management tools in operation indicates the need for continuous refinement of network monitoring policies.

Despite this UDP burst, it can be said that basically every kind of malicious traffic is much more common in traffic coming from the main Internet. Already on a very high level analysis, incoming network scans were evident when analysing distinct IP addresses seen. There are about three times more rejected connections observed among inbound connections, with a majority of them being unreplied scanning attempts, but also a substantial number of immediate reset terminations. Around 90% of the counted header anomalies appeared on the incoming link, which goes hand in hand with the above mentioned scans. These packet header anomalies include inconsistencies in the IP flags, TCP header length and TCP connection flags field, which was discussed in more detail in an earlier study on the MonNet data [10]. Even though these header anomalies are very rare compared to the total number of packets, they indicated again skewed distribution of malicious traffic towards incoming traffic. The inconsistencies were shown to stem from network attacks trying to exploit protocol vulnerabilities as well as active OS fingerprinting tools.

Also the analysis of statistical connection properties within established connections revealed a large number of inbound login attempts to SSH, SMTP or MS SQL servers. Finally, on UDP level scanning traffic and security exploits were shown to happen in more than 90% of the cases within incoming traffic, which are as well in the order of millions in absolute numbers.

This summary of malicious behaviour confirms the suspicion that the main number of anomalies indeed originates on the outside, on the ”unfriendly” Internet. It was shown that anomalies are between 3 and 9 times more common among inbound data. Typical University campus networks, but even student networks, are comparable well behaving, probably due to higher configuration and administration efforts.

7.2. P2P traffic

Except the directional differences due to malicious traffic, P2P is a second source heavily influencing traffic properties. Even with a simple, underestimating port analysis, we could show that P2P traffic is a major part of the traffic, responsible for at least twice as much packets and volume among inbound traffic as compared to outbound traffic. Artefacts of P2P traffic were found in packet size distribution, TCP connection termination behaviour, TCP options and statistical connection properties. P2P traffic was also shown to be a major source for long-duration traces, especially among inbound connections. Additionally, P2P overlay traffic is responsible for the major amount of UDP flows, carrying typically less than 3 small sized packets, but being responsible for several millions of distinct IP addresses observed in the traffic. These short flows are furthermore hard to distinguish from malicious scanning or attacking traffic, which needs to be taken into consideration by network engineers and security experts working on sampled flow level analysis.

7.3. Conclusion

While some high-level analysis, like cumulated traffic volumes or protocol breakdown, could suggest an even distribution between inbound and outbound traffic, this study reveals that there are a number of significant directional differences found on different protocol levels. Especially the detailed TCP connection analysis, contrasting incoming and outgoing established connections by statistical means, revealed significant differences. Even though connections established in both directions show a typical client-server pattern, this behaviour is more pronounced among inbound connections. Generally, inbound connections, established from the outside, are shown to be more likely to carry larger volumes of data (elephants), larger number of packets and experience longer connection lifetimes. However, these differences, caused by the imbalance in P2P traffic, cancel out on high-level summaries because established inbound connections are on the other hand about 10% fewer than outbound connections.

rst of all, the comprehensive analysis yielded required insights for network developers and traffic engineers. Furthermore, the results can be important input in order to improve quality and authenticity of future simulation models. Finally, the highlighted traffic anomalies are relevant for better understanding of security related issues like intrusion detection or detection of large scale attacks.

Acknowledgement

The authors want to thank Pierre Kleberger for his kind technical support and Tomas Olovsson for his valuable and constructive comments throughout the MonNet project.

References

[1] M. Arlitt and C. Williamson, "An analysis of TCP reset behaviour on the Internet," Computer Comm. Review, vol. 35, 2005.
[2] N. Brownlee and K. C. Claffy, "Understanding Internet traffic streams: dragonflies and tortoises," IEEE Communications Magazine, vol. 40, pp. 110-17, 2002.
[3] N. Brownlee and K. C. Claffy, "Internet Measurement," IEEE Internet Computing, vol. 8, pp. 30-33, 2004.
[4] CiscoSystems, "IPsec VPN WAN Design Overview," Cisco Documentation, 2006.
[5] S. Donnelly, "Endace DAG Timestamping Whitepaper," Endace Withepapers, 2006.
[6] S. Floyd and E. Kohler, "Internet research needs better models," Computer Communication Review, vol. 33, pp. 29-34, 2003.
[7] C. Fraleigh, S. Moon, B. Lyles, C. Cotton, M. Khan, D. Moll, R. Rockell, T. Seely, and S. C. Diot, "Packet-level traffic measurements from the Sprint IP backbone," Network, IEEE, vol. 17, pp. 6-16, 2003.
[8] A. Gerber, J. Houle, H. Nguyen, M. Roughan, and S. Sen, "P2P The Gorilla in the Cable," in National Cable & Telecommunications Association(NCTA) National Show. Chicago, IL, 2003.
[9] A. Householder, K. Houle, and C. Dougherty, "Computer attack trends challenge Internet security," Computer, vol. 35, 2002.
[10] W. John and S. Tafvelin, "Analysis of Internet Backbone Traffic with focus on Header Anomalies," submitted for publication, Chalmers, Göteborg, Sweden, 2007.
[11] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, "Transport layer identification of P2P traffic," Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, Taormina, Sicily, Italy, 2004.
[12] M.-S. Kim, Y. J. Won, and J. W. Hong, "Characteristic analysis of internet traffic from the perspective of flows," Computer Communications, vol. 29, pp. 1639-1652, 2006.
[13] K. Lan and A. Hussain, "The Effect of Malicious Traffic on the Network," Proceedings of the Workshop on Passive and Active Measurements (PAM), 2003.
[14] K.-C. Lan and J. Heidemann, "Rapid model parameterization from traffic measurements," ACM Transactions on Modeling and Computer Simulation, vol. 12, pp. 201-29, 2002.
[15] K.-C. Lan and J. Heidemann, "A measurement study of correlations of Internet flow characteristics," Computer Networks, vol. 50, pp. 46-62, 2006.
[16] S. McCreary and K. C. Claffy, "Trends in wide area IP traffic patterns - A view from Ames Internet Exchange," Cooperative Association for Internet Data Analysis - CAIDA, San Diego Supercomputer Center, San Diego 2000.
[17] M. Mellia, R. Lo Cigno, and F. Neri, "Measuring IP and TCP behavior on edge nodes with Tstat," Computer Networks, vol. 47, pp. 1-21, 2005.
[18] A. W. Moore and K. Papagiannaki, "Toward the Accurate Identification of Network Applications," Lecture Notes in Computer Science, pp. 41-54, 2005.
[19] T. Mori, M. Uchida, and S. Goto, "Flow analysis of internet traffic: World wide web versus peer-to-peer," Systems and Computers in Japan, vol. 36, pp. 70-81, 2005.
[20] T. Mori, M. Uchida, R. Kawahara, J. Pan, and S. Goto, "Identifying elephant flows through periodically sampled packets," Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, Taormina, Sicily, Italy, 2004.
[21] K. Pentikousis and H. Badr, "Quantifying the deployment of TCP options - a comparative study," IEEE Communications Letters, vol. 8, pp. 647-9, 2004.
[22] S. Saroiu, K. P. Gummadi, R. J. Dunn, S. D. Gribble, and H. M. Levy, "An analysis of internet content delivery systems," Proceedings of the 5th symposium on Operating systems design and implementation, Boston, Massachusetts, 2002.
[23] C. Shannon, D. Moore, and K. C. Claffy, "Beyond folklore: observations on fragmented traffic," IEEE/ACM Transactions on Networking, vol. 10, pp. 709-20, 2002.
[24] S. Tafvelin, "Presentation: QoS measurements," TERENA Networking Conference, Poznan, Poland, 2005.
[25] K. Thompson, G. J. Miller, and R. Wilder, "Wide-area Internet traffic patterns and characteristics," IEEE Network, vol. 11, 1997.
[26] P. A. Watson, "Slipping in the Window: TCP Reset Attacks," Technical Whitepaper, 2003.
[27] J. Xu, J. Fan, M. Ammar, and S. B. Moon, "On the design and performance of prefix-preserving IP traffic trace anonymization," Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement, San Francisco, California, USA, 2001.
[28] Y. Zhang, L. Breslau, V. Paxson, and S. Shenker, "On the characteristics and origins of Internet flow rates," Computer Communication Review, vol. 32, pp. 309-322, 2002.

PAPER III

Wolfgang John and Sven Tafvelin

Heuristics to classify Internet Backbone Traffic based on Connection Patterns

ICOIN '08: 22nd International Conference on Information Networking
(Proceedings published by the IEEE Communications Society)
Busan, Korea, 2008

Heuristics to Classify Internet Backbone Traffic based on Connection Patterns

Wolfgang John and Sven Tafvelin

Department of Computer Science and Engieneering
Chalmers University of Technolgy
Göteborg, Sweden
{johnwolf,tafvelin}@chalmers.se

***Abstract*— In this paper Internet backbone traffic is classified on transport layer according to network applications. Classification is done by a set of heuristics inspired by two previous articles and refined in order to better reflect a rough and highly aggregated backbone environment. Obvious misclassified flows by the existing two approaches are revealed and updated heuristics are presented, excluding the revealed false positives, but including missed P2P streams. The proposed set of heuristics is intended to provide researchers and network operators with a relatively simple and fast method to get insight into the type of data carried by their links. A complete application classification can be provided even for short 'snapshot' traces, including identification of attack and malicious traffic. The usefulness of the heuristics is finally shown on a large dataset of backbone traffic, where in the best case only 0.2% of the data is left unclassified.** [1]

I. Introduction

Reliable classification of Internet traffic based on network applications is still an open research issue. However, network operators need to know the type of traffic they are carrying, amongst others in order to improve network design and provisioning and to support QoS and security monitoring. Ongoing measurements will furthermore reveal trends and changes in the usage of network applications. A good example is the shift in the early 2000's, when P2P file sharing replaced HTTP as the Internet's 'killer application', implying not only changes in data volumes, but also in traffic properties.

Different approaches to classify network traffic exist. Traditionally, traffic was classified based on *source and destination port numbers*. While this approach is very simple and does not require any packet payload, it is highly unreliable in modern networks. This is especially true for most P2P applications, which are trying to disguise their traffic in order to evade traffic filters and legal implications. It was shown that pure port number analysis underestimates actual P2P traffic volumes by factors of 2 to 3 [1].

A more reliable technique involves analysis of *packet payloads*. This approach can potentially provide highly accurate results given a complete set of payload signatures [2]. Beside the high effort of keeping the set of signatures updated, this method relies on network traces including packet data, which is uncommon due to privacy and legal concerns. Furthermore matching payload signatures on high-speed links is far from trivial and poses high processing requirements.

[1]This work was supported by SUNET, the Swedish University Network

A more recent classification technique is based on *statistical properties* of flows. A promising feature of these methods is that they are neither relying on port numbers nor on packet payload. However, the success of such 'statistical fingerprints' highly depends on the accuracy of the training data used. Ensuring accuracy and authenticity of the training sets is still an open issue [3], especially for disguised P2P flows.

Finally, network data can be classified according to *connection patterns*. Instead of looking at individual packets or flows, sequences of flows to or from a specific endpoint are matched with a set of predefined heuristics [4], [5]. These heuristics typically don't require packet payload and could potentially even disregard port numbers.

We initially intended to classify Internet backbone data in order to investigate the influence of P2P applications on traffic properties. Consequently it was planned to apply an existing and verified classification technique. Since our available datasets did not include packet payload and accurate training data, payload signatures or statistical fingerprinting could not be applied. Thus applying straight-forward connection pattern heuristics was the obvious approach. In [4], Karagiannis presents a set of two heuristics for transport layer identification of P2P traffic, including seven rules for removing false positives. The paper verifies that their method can identify 95% of P2P flows, with around 10% false positives compared to a carefully carried out payload analysis on OC-48 backbone data. Additionally, Perenyi [5] recently proposed an updated set of six heuristics to identify and analyze P2P traffic, based on very similar ideas like Karagiannis. These heuristics were verified against traffic generated in a lab environment, yielding a hit ratio for P2P traffic of over 99%, with less than 1% false positives or unclassified P2P flows.

After applying the approaches of both Karagiannis and Perenyi to our data, it turned out that their results differ substantially. Furthermore, obvious false positives were detected in our data with both classification methods. As a result, we propose a refined combination of the heuristics by Karagiannis and Perenyi including some additions. The modifications were necessary to make the classification suitable for relatively short traces of a harsh Internet backbone environment, including highly aggregated and diverse traffic with a substantial amount of attacking and malicious traffic. Besides being based on the verified heuristics of Karagiannis and Perenyi, the results

where further verified by manual inspection. Flows, which are not classified as P2P traffic by all three applied sets of heuristics are separately discussed regarding their most probable traffic class, thereby identifying obvious misclassification.

II. Data Description

Our dataset was collected during 20 days in April 2006 on the OC192 backbone of the Swedish University Network (SUNET). During this period, four traces of 20 minutes were collected each day at identical times (2AM, 10AM, 2PM, 8PM), as described in [6] and [7]. After recording the packet level traces on the 2x10 Gbit/s links, payload beyond transport layer was removed and IP addresses were anonymized due to privacy concerns. A per-flow analysis was conducted on the resulting bidirectional traces, where flows are defined by the 5-tuple of source and destination IP and port numbers as well as transport protocol. TCP flows represent connections, and are therefore further separated by SYN, FIN and RST packets. UDP flows are separated by a timeout of 64 seconds. The 73 traces in the dataset sum up to 10.7 billion packets, containing 7.5 TB of data. We identified 81 Million TCP connections and 91 Million UDP flows, with the TCP connections carrying 97% of all data. The further analysis is dealing with TCP connections only, even though the classification heuristics have been successfully applied to UDP flows as well.

III. Proposed Heuristics

The set of heuristics proposed in this paper is strongly inspired by the heuristics by Karagiannis [4] and Perenyi [5], and will therefore be presented briefly only. The classification is based on connection patterns, but in some cases also port numbers are taken into account. Besides the rules for filtering out P2P traffic (H1-H5), a number of heuristics are used to remove false positives from flows suspected to be P2P traffic (F1-F10). These 'false positive' rules in turn can be used to classify other types of traffic, as shown in section V. In contrast to Perenyi's approach, most of our proposed heuristics (with exception of H5 and F10) are first applied independently to all flows and are then prioritized. We apply these heuristics to our dataset in 10 minute intervals, which means that every interval is analyzed self-contained, without memory of previous intervals. Even though such memory could improve the accuracy of the results, our approach has the advantage to allow operators to classify snapshots of their traffic fast and in an ad hoc fashion. We will show that even 10 minute intervals can provide satisfying results. The proposed heuristics include a number of thresholds which might be adjusted. For our data the thresholds used were derived empirically through experiments on a number of traces. In the following list of heuristics, *(K)* (Karagiannis) or *(P)* (Perenyi) indicate by which previous method the heuristic was inspired, while *(J)* (John) marks newly introduced rules.

H1: TCP/UDP IP Pairs:(K),(P). This rule exploits the fact that many P2P applications use TCP for data transfer and UDP for signaling traffic. Source and destination IP pairs, which concurrently use TCP and UDP are therefore marked as P2P hosts. All flows to and from these hosts are marked as potential P2P flows. Concurrent here means usage of TCP and UDP within the 10 minutes interval. Karagiannis identified some non-P2P applications which show a similar behavior, such as netbios, dns, ntp and irc (Table 3 in [4]). UDP flows from these applications are excluded from this heuristic based on their port-numbers.

H2: P2P Ports:(P). Even though many P2P applications choose arbitrary ports for their communication, approx. one third of all P2P traffic can still be identified by known P2P destination port numbers [1]. Furthermore, it seems disadvantageous for non-P2P applications to deliberately use well known P2P ports for their services, since traffic on these ports is often blocked by traffic filters in some networks. Flows to and from port numbers listed in Table 3 of [5], enriched with additional P2P ports, are marked as potential P2P traffic.

H3: Port Usage:(P). In normal application, the operating system assigns ephemeral port numbers to source ports when initiating connections. These numbers are often iterating through a configured ephemeral port space. It is very unusual, that the same port numbers are used within short time periods. This however can be the case for P2P applications with fixed ports assigned for signaling traffic or data transfer. If a source port on a host is repeatedly used within 60 seconds, the host is marked as P2P host, and all flows to and from this host are marked as potential P2P flows.

H4: P2P IP/Port Pairs:(K). If listening ports on peers in P2P networks are not well known in advance, they are typically propagated to other peers by some kind of signaling traffic (e.g. an overlay network). This means that each host connecting to such a peer will connect to this agreed port number, using a random, ephemeral source port. As noted by Karagiannis, P2P peers usually maintain only one connection to other peers, which means that each endpoint (IP,port) has at least the same number of distinct IP addresses (#sIP) and number of distinct ports (#sPort) connected to it. If #sPort-#sIP<2 and #sIP>5, the host is considered as P2P host, and all flows to and from this host are marked as potential P2P.

F1: Web IP/Port Pairs:(K). Web traffic on the other hand typically uses multiple connections to one server. For this reason hosts are marked as web-hosts, if the difference between #sPort and #sIP connected to an endpoint (IP,port) is larger than 10, the ratio between #sPort and #sIP is larger than two and at least 10 different IPs are connected to this endpoint (#sPort-#sIP>10 and #sPort/#sIP>2 and #sIP>10). All flows with http port numbers (80, 443, 8080) to and from these webhosts are then marked as web traffic.

F2: Web:(P). To further identify web traffic, we follow Perenyi's heuristic number 2, taking advantage of the fact that web clients typically not only use multiple, but even parallel connections to webservers. Hosts with parallel connections to a http port are considered as webservers. All flows to and from web servers on http ports are marked as web traffic.

F3: DNS:(K). Traditional services like dns sometimes use equal source port and destination port numbers. As suggested by Kargiannis, we mark endpoints (IP,port) as non-P2P, if it includes flows with equal source- and destination port and port numbers smaller than 501. All flows to and from this endpoint are then marked as non-P2P traffic.

F4: Mail:(K). Hosts receiving traffic on mail ports (smtp, pop, imap) and in the same analysis interval also initiate connections to port 25 on other hosts are considered to be mailservers. All flows to and from mailservers are marked as mail traffic.

F5: Messenger:(K). Popular messenger and chat servers (icq, yahoo, msn, jabber, irc) tend to have long uptimes and rarely change IP addresses, especially when maintained by commercial providers such as Microsoft and Yahoo. To improve the accuracy of the results, in this heuristic we therefore take advantage of the whole 20 day long dataset. Hosts, connected to by at least 10 different IPs on well known messenger ports within a period of at least 10 days, are marked as messenger servers. All traffic to and from these hosts on known messenger ports is classified as messenger traffic.

F6: Gaming:(J). Popular game servers (currently only the most common online games Half-Life and World of Warcraft) are identified in the same fashion as messenger servers. All traffic to and from the game servers on well known gaming ports is classified as gaming traffic.

F7: Ftp: (J). Ftp was not taken into account by Karagiannis, while Perenyi implicitly included it as part of its 'well known port' rule. Identifying data transfer in passive ftp remains a problem. Active ftp data transfer on the other hand can easily be marked as ftp traffic identified by an initiating sourceport number of 20, as used by ftp servers to actively serve their requesting clients.

F8: non P2P Ports:(P). As noted by Perenyi, destination ports are still suitable to identify traffic of some common applications. Our set of well known non-P2P ports includes netbios, dns, telnet, ssh, ftp, mail, rtp and bgp. All flows to the listed destination ports are marked as non-P2P flows.

F9: Attacks:(J). This rule is probably the most significant improvement to the original heuristics. While Perenyi does not take malicious traffic into account at all, Karagiannis rules out simple network scans as false positives. We first identify suspicious pairs of source IPs and destination Ports (*AttackPairs*). All flows with source IP and destination port inside the list of AttackPairs are then marked as attacks. AttackPairs are identified by three different cases:

a) *Sweep*: The ratio between number of destination IPs (#dIP) and number of destination ports (#dPort) from a certain host is greater than 30. This means that one host is connecting to a lot of hosts with only a few different port numbers, as typically the case when scanning IP ranges for vulnerabilities on specific ports.

b) *Scan*: The ratio between #dIP and #dPort is less than 0.33 and #dIP is less than 5. This would be the case if one host is scanning a small number of specific, dedicated targets on a large number of different ports.

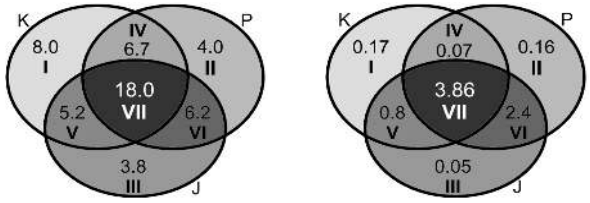
c) *DoS*: #dIP is less than 5, #dPort is less than 5 and the average number of conn. per sec (conn/s) is greater than 6. This behavior represents 'hammering' attacks, where one host is trying to overload a few targets (typically one) by opening connections to a few services very frequently.

F10: unclassified, known non-P2P Port:(J). Up to this point all heuristics mark flows independent of each other. All flows left unmarked until now are neither suspected to be P2P traffic nor obvious cases of non-P2P traffic. We believe it is safe now to apply a port number classification on the previously unclassified flows. All flows, whose source- or destination port number matches a set of well-known non-P2P port numbers including (http, messenger, game) are classified non-P2P, if not classified by any heuristics (H1-H4, F1-F9).

H5: unclassified, long flow:(P) After removing well known applications from the unclassified flows, we mark remaining unclassified flows which carry more than 1 MB of data in one direction or have connection durations of over 10 minutes as P2P flows. This rule is based on Perenyis heuristic 6, even though we believe it is a very weak rule. However, there is a large probability, that such long flows in fact are P2P flows.

After running an analysis on our dataset based on the presented heuristics, we classify all flows as P2P traffic which have been classified by one or more of the heuristics H1-H5, and at the same time NOT being classified by any of the false positive heuristics F1-F10. In Section IV, flows marked by H5 are included to P2P traffic. However, in Section V we chose to treat traffic classified by this heuristic separately.

Weaknesses: The above suggested mixture of connection pattern and port number classification has some weaknesses. First of all, the analysis interval can greatly influence the success of the heuristics, especially for those analyzing connection patterns. Longer intervals yield better results given that the various empirical thresholds are adjusted. A natural border for the analysis interval is obviously given by memory and computational constraints. Additionally, there is a risk with too long intervals since activities on the Internet are often short lived, and e.g. a host doing a scanning campaign on port 80 might simply surf the Internet an hour later. Another problem in this context are networks behind NATs or with dynamically assigned IP addresses. A second weakness is the length of the traces used. For connections established before the measurement interval the initiator is unknown, and it is unclear which host is source and which is destination. Additionally there is typically some asymmetrically routed traffic in backbone networks, which needs to be considered as special case when implementing the heuristics. Furthermore, heuristics based on connection patterns are depending on a certain amount of connections per host during the analysis interval. Finally, heuristics relying on empirical thresholds are not fail-proof, and it is possible to come up with examples for false positives for any of them. However, both Karagiannis and Perenyi proved that these heuristics can be effective when carefully prioritizing the different rules.



(a) #connections (10^6) (b) Amount of data (TB)

Fig. 1. P2P traffic by Karagiannis (K), Perenyi (P) and new proposal (J)

IV. Verification of the proposed Heuristics

To verify the proposed adjustments, we classified our backbone data by each of the three sets of heuristics (Karagiannis, Perenyi and our own proposal in section III). For each flow, a bitmask was set in a database according to matching rules. This method allowed us to analyze intersections between the three approaches separately - meaning flows marked as P2P traffic by either one, two or all three of the approaches. The results are illustrated by the Venn diagrams in fig.1, presenting connection counts (a) and amount of data (b) in absolute numbers. The three circles represent P2P flows classified by the different rule-sets (Karagiannis left, Perenyi right, new proposal beneath). The following paragraphs will discuss the different intersections (IS I-VII), thereby motivating the proposed modifications and additions to the original approaches.

IS I: This intersection represents flows classified as P2P by Karagiannis only. A number of updated rules identified these flows as false positives. Rule F9 (attacks) marked 53% of them, often classified as known non-P2P ports by Perenyi. This is plausible, considering that these connections are mainly 1-packet flows, directed to popular scanning ports (135, 139, 445). Rule F2 (web) classified another 25% of these connections, carrying 40% of the data in this intersection. Since parallel connections to http ports are a strong indication for web traffic, F2 is regarded as a reliable rule. F8 (non P2P-ports) accounts for 15% of these connections, carrying 43% of the data, mainly on ports for rtp, ssh and mail. This is plausible, since it is common that these applications carry large amounts of data, so there is no reason considering them as P2P flows. The remaining flows are either marked by F7 (active ftp) or F10 (unclassified, but known non-P2P port).

IS II: In this intersection, 99% of the data was classified as P2P by Perenyi's 'long flow' rule only. This is obviously Perenyi's weakest heuristic, since it simply considers any flow carrying more than 1 MB of data or lasting longer than 10 minutes as P2P. 75% of this data is considered as false positive according to F10. Unclassified by any other heuristic, a pure port number classification marks these flows as web flows according to their destination http ports. Another 10% are marked as web traffic by F2. The remaining data was classified by F4 (mail), F5 (messenger) and F6 (gaming), all three considered to be accurate rules, taking connection patterns and port number into account. In terms of connection numbers, 95% of the connections in IS II are again identified as false positives by F9 (attacks) with similar properties as in IS I.

IS III: All of the flows only classified as P2P by the proposed heuristics are unclassified by Perenyi. Even Karagiannis left 45% unclassified, with the remaining 45% classified by the non-P2P IP/Port Pair rule. In [4] this rule was identified as unreliable if less than 5 IPs are connected to an IP/Port Pair. Since in H4 this restriction was taken into account, it is plausible to include the flows marked as P2P in IS III based on combinations of H4 and/or H3 (port usage).

IS IV: The flows classified as P2P by both Karagiannis and Perenyi are in 98% of the cases again marked as false positives by F9 (attacks), carrying very little data. In terms of data, Perenyi's 'long flow' rule and Karagiannis' IP/Port Pair rule are responsible for 90% of the data in this intersection. As discussed above, both rules are considered rather weak. Since additionally none of the refined P2P heuristics (H1-H4) matched, rule F10 (unclassified, but well known port) is reason enough to exclude 80% of this flows as false positives (mainly targeting http ports). The remaining flows have been marked by F1 (web pairs), F5 (messenger) and F6 (gaming).

IS V: In this intersection, flows are entirely unclassified by Perenyi. Since these flows are classified as P2P by both Karagiannis and the proposed heuristics, there is no reason not to consider them as P2P traffic.

IS VI: Perenyi's 'long flow' rule identified 77% of the data in this large intersection as P2P, with the remaining connections classified according to known P2P port numbers. The proposed heuristics on the other hand classify 88% of these flows as P2P by H2-H4, accounting for 72% of data. Most of the data is even classified by 2 or 3 of the heuristics. The remainder (685 GB) is classified by H5 (long flows) only, and will therefore be treated as a special category in our results section. Karagiannis leaves a large part (60%) of this intersection unclassified, with the rest classified by the non-P2P IP/Port Pair rule, which is an inaccurate rule for endpoints with few connected hosts as noted above. Since there is no strong indication to rule out flows as false positives, they are classified as P2P except for the 685 GB by H5 (long flows).

IS VII: Data in this intersection is classified as P2P by both Karagiannis and Perenyi, and no false positives were identified by the proposed heuristics. Consequently, there is no reason not to consider this intersection as P2P.

V. Classification Results

We finally applied the proposed heuristics to our data traces (Section II). Fig.2 represents time series of classified network protocols. The x-axis of the graphs represents time, with one bar for each trace time (2AM, 10AM, 2PM and 8PM). Four traces on three days (07/04, 09/04, 23/04) had to be discarded due to measurement errors. The remaining whitespaces between bars represent the 8 hour measurement break between 2AM and 10AM, which means that each continuous block represents 4 traces collected in the order of [10AM, 2PM, 8PM, 2AM]. The first graph shows total amount

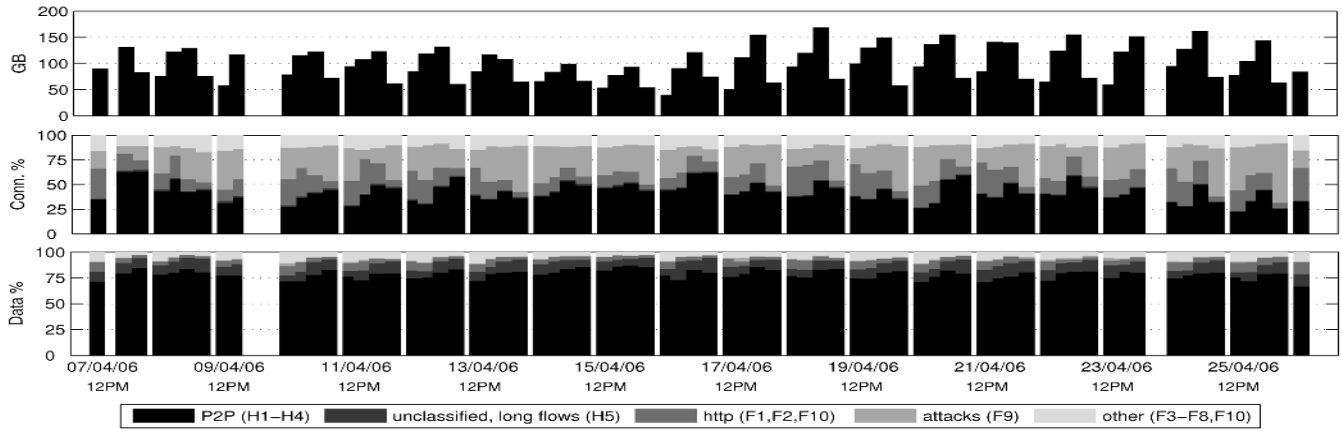


Fig. 2. TCP data vs trace times (first row); Appl. breakdown by #conn. (second row); Appl. breakdown by data carried (third row)

of TCP data in GByte versus trace times. The second and third row illustrate application breakdown for the particular trace in terms of connection numbers and data volumes.

In the connection breakdown, only four categories are visible, since flows classified by H5 are too small in number to show up in this graph. Anyhow, these 31,000 long flows are responsible for almost 10% of the TCP data. Typically, these flows begin and end outside the measurement period and transfer data between hosts, which do not generate additional traffic on our links. Since our classification method is based on connection patterns, insufficient connection numbers for a particular host reveal a weakness of this method. In the data breakdown on the other hand, flows classified by F9 (attacks) are not visible. Even though attacks represent between 8 and 60% of the flows, they carry less than 1% of the data on average. This also proves the power of F9, since it effectively detects DoS attacks and network scanning, which typically show up as short 1-packet flows only, carrying no payload data. P2P flows (flows matching H1-H4, while not matching any of the false positive rules F1-F10) account for an average of 42% of the connections. On the other hand, they carry between 66 and 87% of the traffic, with an average of 79%. This indicates once more the success of the heuristics, since P2P flows are expected to carry more data on average than non-P2P flows. On this dataset, the proposed heuristics left as little as 1% of the connections and 0.2% of the data unclassified (except the flows classified by H5).

While a careful analysis of these results need to be done as future work, the short result section should indicate the power and usefulness of the proposed heuristics.

VI. Summary and Conclusions

This article proposes a set of heuristics for classifying backbone-type data according to applications. The proposed heuristics are intended to provide researchers and network operators with a comparably simple[2] method to get insight into the type of data carried by their links. Furthermore these heuristics work on traces as short as 10 minutes, which allows operators to classify snapshots of their traffic relatively fast, by only adjusting applied thresholds and parameters empirically. The heuristics can be used to classify backbone traffic according to a number of applications, including P2P traffic, web traffic and other common applications. Furthermore, we introduce a new rule that successfully identifies network attacks, which is an additional feature for network operators and researchers interested in network security or intrusion detection issues. Some of the proposed heuristics are based on two existing methods. Besides relying on the verification methods of these original heuristics, a careful analysis of the resulting classifications was carried out, pinpointing obvious cases of false positives. Both previous sets of heuristics overestimate the number of P2P flows, mainly because attacking traffic is not taken into account accordingly. On the other hand, both methods underestimate the amount of P2P data on the links. By combining the successful rules of the two methods and adding new, necessary rules, a set of refined and updated heuristics is presented. The heuristics are successfully applied to a large collection of backbone data, yielding a valuable breakdown of applied application protocols. When considering the few large flows classified by the H5 rule as P2P traffic, the proposed heuristics leave only 0.2% of the data unclassified.

[2]Simple, because it does not require packet payloads, updated payload signatures or training data for statistical fingerprinting methods.

References

[1] A. W. Moore and K. Papagiannaki, *Toward the Accurate Identification of Network Applications*, ser. Lecture Notes in Computer Science, 2005.

[2] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of p2p traffic using application signatures," ser. 13th International World Wide Web Conference, New York, USA, 2004.

[3] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli, "Traffic classification through simple statistical fingerprinting," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 1, pp. 5–16, 2007.

[4] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, "Transport layer identification of p2p traffic," in *Proceedings of the 4th ACM Conference on Internet Measurement*, Taormina, Sicily, Italy, 2004.

[5] M. Perenyi, D. Trang Dinh, A. Gefferth, and S. Molnar, "Identification and analysis of peer-to-peer traffic," *Journal of Communications*, vol. 1, no. 7, pp. 36–46, 2006.

[6] W. John and S. Tafvelin, "Analysis of internet backbone traffic and header anomalies observed," in *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, San Diego, California, USA, 2007.

[7] ——, "SUNET OC 192 Traces, April 2006 (collection)," http://imdc.datcat.org/collection/1-04HN-W=SUNET+OC+192+Traces%2C+April+2006 (accessed 071207).

PAPER IV

Wolfgang John, Sven Tafvelin and Tomas Olovsson

Trends and Differences in Connection Behavior within Classes of Internet Backbone Traffic

accepted for presentation at

PAM '08: the 9th Passive and Active Measurement Conference

(Proceedings to be published in the Springer Lecture Notes in Computer Science)

Cleveland, Ohio, USA, 2008

Trends and Differences in Connection-behavior within Classes of Internet Backbone Traffic

Wolfgang John, Sven Tafvelin, and Tomas Olovsson

Department of Computer Science and Engineering
Chalmers University of Technology, Göteborg, Sweden
Email: {johnwolf,tafvelin,tomas}@chalmers.se

Abstract. In order to reveal the influence of different traffic classes on the Internet, backbone traffic was collected within an eight month period on backbone links of the Swedish University Network (SUNET). The collected data was then classified according to network application. In this study, three traffic classes (P2P, Web and malicious) are compared in terms of traffic volumes and signaling behavior. Furthermore, longitudinal trends and diurnal differences are highlighted. It is shown that traffic volumes are increasing considerably, with P2P-traffic clearly dominating. In contrast, the amount of malicious and attack traffic remains constant, even not exhibiting diurnal patterns. Next, P2P and Web traffic are shown to differ significantly in connection establishment and termination behavior. Finally, an analysis of TCP option usage revealed that Selective Acknowledgment (SACK), even though deployed by most web-clients, is still neglected by a number of popular web-servers.[1]

1 Introduction

Today, many network operators do not know which type of traffic they are carrying. This problem emerged mainly in the early 2000's, when P2P file sharing applications started to disguise their traffic in order to evade traffic filters and legal implications. Since then, the network research community started to draw increasing attention to classification of Internet traffic. Traditional port number classification was shown to underestimate actual P2P traffic volumes by factors of 2-3 [1], thus more sophisticated classification methods have been proposed. These methods are typically either based on payload signatures [2], statistical properties of flows [3] or connection patterns [4].
A number of articles also present properties of different traffic classes resulting from traffic classification. Gerber et al. [5] classified flow measurements from a tier-1 ISP backbone in 2003. Even if their classification method has been based on port numbers, they indicate a dominance of P2P applications. Sen et al. [6] investigated connectivity aspects of P2P traffic on different levels of aggregation (IP, prefix, AS) in 2002. The study was based on flow data collected at a single ISP, classified by a port number method. More recent articles from 2005 and 2006

[1] This work was supported by SUNET, the Swedish University Network

present differences between P2P and non-P2P traffic in terms of flow properties such as size, duration and inter-arrival times [7, 8]. Perenyi et al. [8] additionally presents a comparison of diurnal patterns for P2P vs. non-P2P traffic.
This article presents the results of a classification of current Internet backbone data. The datasets do not include packet payloads, thus connection pattern heuristics [9] were used to classify the datasets. The classification approach, disregarding packet payload data, has the advantage of avoiding legal issues and has the capability to classify even encrypted traffic, which is gaining popularity among P2P traffic. We chose to focus on 3 main traffic classes: (1) P2P file sharing protocols; (2) Web traffic; (3) malicious and attack traffic. First, we show how these traffic classes develop over a time period of eight months by highlighting trends in traffic volumes and connection numbers, also pointing out some diurnal differences. Next, we present differences between the traffic classes in terms of connection signaling behavior. This includes success rates for TCP connection establishment, a breakdown of different TCP connection termination possibilities and TCP option usage within established connections.
To our knowledge, this is the first attempt to characterize differences and trends within traffic classes in terms of connection signaling, with exception of a brief discussion about connection termination in [10]. We provide a thorough analysis of differences and trends for the selected traffic classes, since they have a major impact on the overall traffic behavior on the Internet. It is of general importance to follow trends in contemporary Internet traffic in order to react accordingly in both infrastructure and protocol development. Furthermore a thorough analysis of specific connection properties reveals how different traffic classes are behaving 'in the wild'. Since the data analyzed was collected on a highly aggregated backbone during a substantial time period, the results reflect contemporary traffic behavior of one part of the Internet. These results are thereby not only valuable input for simulation models, they are also interesting for developers of network infrastructure, applications and protocols.

2 Data Description

The two datasets used in this article [11] were collected in April (spring dataset) and in the time from September to November 2006 (fall dataset) on an OC192 backbone link of the Swedish University Network (SUNET). In spring, four traces of 20 minutes were collected each day at identical times (2AM, 10AM, 2PM, 8PM) as described in [12]. The fall dataset was collected at 276 randomized times during 80 days. At each random time, a trace of 10 minutes of duration was stored. To avoid bias when comparing the datasets, we treated the 20 minute samples from spring as two separate 10 minute traces. Furthermore, for this study traces from fall are only considered if collected during the time-window between 20 minutes prior and after the collection times of spring (e.g. 1:40AM-2:40AM). When recording the packet level traces on the 2x10GB links, payload beyond transport layer was removed and IP addresses were anonymized due to privacy concerns. After further pre-processing of the traces, as described in [11] and

[12], a per-flow analysis was conducted on the resulting bi-directional traces. Flows are defined by the 5-tuple of source and destination IP, port numbers and transport protocol (TCP or UDP). TCP flows represent connections, and are therefore further separated by SYN, FIN and RST packets. For UDP flows, a flow timeout of 64 seconds was used [4]. The 146 traces in the spring dataset include 81 million TCP connections and 91 million UDP flows, carrying a total of 7.5 TB of data. The reduced fall dataset, consisting of 65 traces, includes 49 million TCP connections and 70 million UDP flows, carrying 5 TB of data. In both datasets, TCP connections are responsible for 96% of all data.

3 Methodology

The resulting 130 million TCP connections and 161 million UDP flows have been fed into a database, including per-flow information about packet numbers, data volumes, timing, TCP flags and TCP options. The flows have then been classified by use of a set of heuristics based on connection patterns. The classification method was introduced and verified on the April dataset, as described in [9]. The heuristics are intended to provide a relatively fast and simple method to classify traffic, which was shown to work well on traces even as short as 10 minutes. In the present study the flows are summarized into three different traffic classes: P2P (file-sharing); Web or HTTP (incl. HTTPS); Malicious and attack (i.e. scan, sweep and DoS attacks). Remaining traffic was binned in a fourth class, denoted 'others'. 'Others' includes mail, messenger, ftp, gaming, dns, ntp and remaining unclassified traffic. The latter accounts for about 1% of all connections. In this study, the focus is on trends and differences between P2P and Web traffic, with some notable observations from malicious traffic highlighted as well. Besides the traffic classification, an analysis of traffic volumes and signaling properties is carried out in two further dimensions: longitudinal trends between April and November and diurnal patterns between the four time clusters (times of day).

4 Trends in Traffic Volumes

Longitudinal trends in TCP traffic volumes have been analyzed by building time series for the three traffic classes within each of the four time clusters, representing times of day (2AM, 10AM, 2PM, 8PM). Due to space limitations, only a condensed time series of TCP traffic is illustrated in Fig.1. The x-axis of the graphs represent time, with one bar for each 10 minute long trace. The first row indicates an increase in traffic volume during 2006. While peak volume per 10 minutes lies at 70 GB in early April, volume reaches 85 GB in late April (right after Easter vacation). This trend continues, with peaks of 94 GB in September and finally 113 GB in November. During one specific interval on November 8 as much as 131 GB have been transfered via TCP. All peak intervals fall into the time cluster of 8PM. The second busiest time cluster in terms of traffic volumes is the one at 2PM. Transfer volumes during 2PM reach on average 80% of the peak values at 8PM. Nighttime and morning hours (2AM, 10AM) show the

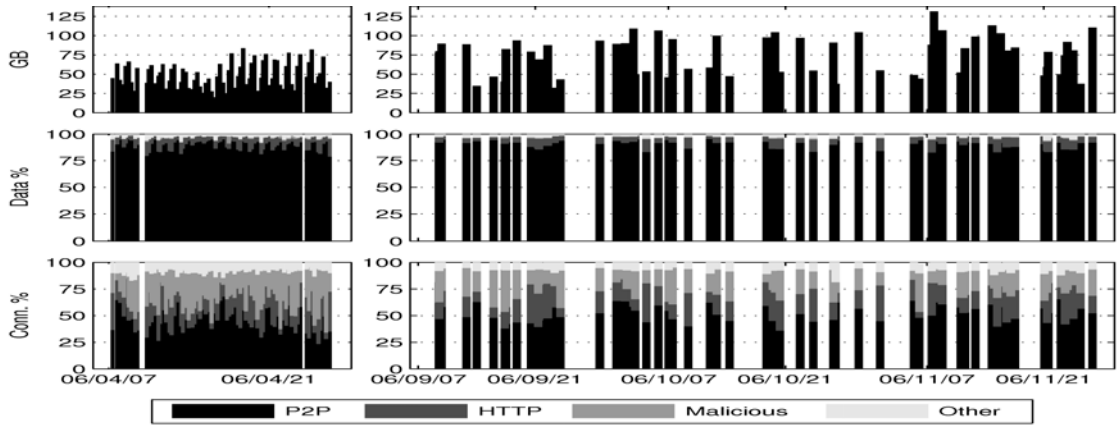


Fig. 1. TCP data vs time (1st row); Appl. breakdown by data(2nd) and #conn.(3rd)

lowest activity with half the transfer volumes of the busy evening hours. This diurnal pattern is best visible in the April section of the first row in Fig.1.

Even if there is an increase in data volumes of around 65% during a time period of eight months, the breakdown into traffic classes remains constant. P2P applications account constantly for as much as 93% and 91% of the data during evening and night time, respectively. During office hours (10AM, 2PM) the fraction of P2P data is reduced to 86%. HTTP, in contrast, is responsible for 9% of TCP data transfered during office hours, and drops down to 5% and 4% during evening and night time. This diurnal difference is explained by a network prefixes analysis, yielding that most P2P traffic originates from student dormitories whereas Web traffic is commonly generated by Universities. The remaining data fractions account mainly for 'other' traffic, since malicious traffic and attacks tend to be single packet flows, not carrying substantial amounts of data.

The traffic breakdown in terms of connection numbers clearly shows that P2P connections typically carry higher amounts of data. Between 40% and 55% of the connections are classified as P2P, following the diurnal patterns of traffic volumes. HTTP connections account for 25% of all TCP connections during office hours, but drop down to 7% at night hours. Interestingly, the fractions of both P2P and HTTP connections (or connection attempts) increased slightly from April to November, while the fraction of malicious traffic decreased from around 30% to 20% during the same time. This development turns out to be a consequence of the constant nature of malicious traffic, such as scanning attacks. In absolute numbers, this traffic class remained remarkably constant during the eight months. Due to the increase in overall traffic volume, its relative fraction evidently was decreased. Since malicious or attack traffic shows neither longitudinal trends nor any significant diurnal pattern, we conclude that this type of traffic rather forms a constant 'background noise' in the Internet.

A similar analysis was also done for UDP flows. Even though larger in number, they are only responsible for 4% of all data. UDP data volumes during 10 minutes increased from peak values of 2.8 GB in April up to 4.6 GB in November.

As in the case of TCP, peak intervals fall into the 8PM time cluster. Afternoon hours experience moderate UDP data volumes, and little UDP activity takes place during night and morning hours.
P2P flows over UDP carry in 76% of all cases less than three packets, which can be explained by signaling traffic as commonly used in P2P overlay networks such as Kademlia. In April, P2P flows are responsible for around 80% of UDP data volumes and connection counts, while the fraction has increased to about 84% in November. In absolute numbers, UDP P2P flow counts have even doubled from April until November, which shows that P2P applications deploying overlay networks via UDP are gaining popularity. Other traffic, including traditional UDP services like NTP or DNS, accounts on average for only 8% of the UDP flows. As for TCP, malicious traffic remains very constant in absolute numbers, which means that relative fractions decreased from 12% to around 8% in November.

5 Differences between Traffic Classes

The following subsection highlights differences between P2P, Web and malicious connections in terms of establishment and termination behavior. In the next subsection, TCP option deployment for P2P and Web connections is compared.

5.1 Differences in Connection Behavior

Fig.2 breaks down the success-rates of connection attempts for the three classes. Established connections include TCP flows with successfully carried out 3-way-handshakes. The second group of connection attempts did not fulfill 3-way-handshakes, but included an initial SYN packet. Finally, there are flows with no SYN seen. These are TCP sessions starting before the measurement interval. Such session fragments account for 13.5% of the 130 million connections seen. Malicious traffic usually consists of 1-packet flows only, which explains why only few malicious connection attempts fall into the no SYN category. In the further analysis, we will only focus on connections including initial SYN packets.
A notable trend can be observed in the P2P graph in Fig.2, where the fraction of unsuccessful connection attempts increased from an average of 49% in April to 54% in November. Web traffic on the other hand has significantly larger fractions

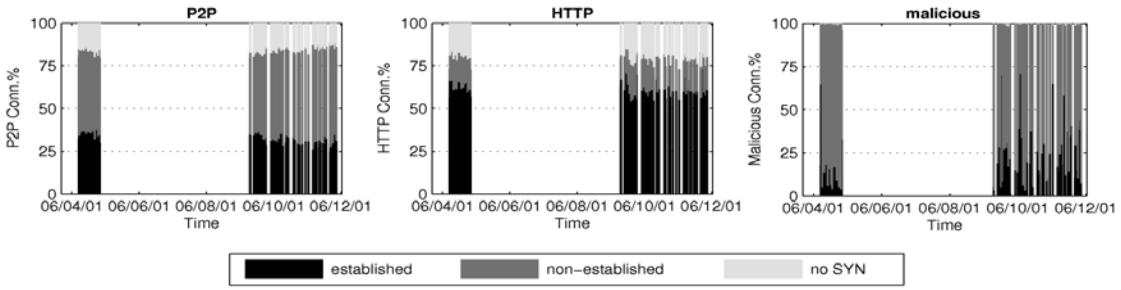


Fig. 2. TCP Connection Breakdown

of established connections, leaving only an average of 16.3% non-established. Malicious traffic is more likely to be established in the fall data, even though a majority of the malicious connections are still connection attempts. The increase in established attack connections is caused by an increase in login attempts to MS-SQL and SSH servers, with a few MS-SQL servers at a local University responsible for the majority of the attempts. According to SANS Internet Storm Center (ISC), malicious activities on both SSH (22) and MS-SQL (1433) ports increased significantly during 2006, which explains the trends seen here.
P2P and malicious connections reveal no diurnal patterns. Within Web traffic however, unsuccessful connection attempts account constantly for around 17.5% during all day, with exception of a drop to 10% during night time hours (2AM). We have no explanation for this phenomena other than HTTP connections are very rare in absolute number during night hours, which makes the statistical analysis more sensitive to behavior of individual applications or user groups.

Non-established connections: Non-established TCP connections have been further divided into connection attempts with one SYN packet only, attempts with direct RST reply and asymmetrical traffic (Fig.3). Due to transit traffic and hot-potato routing, 13% of the connections are asymmetrically routed. Naturally, it is not possible to observe a three-way handshake in this case.

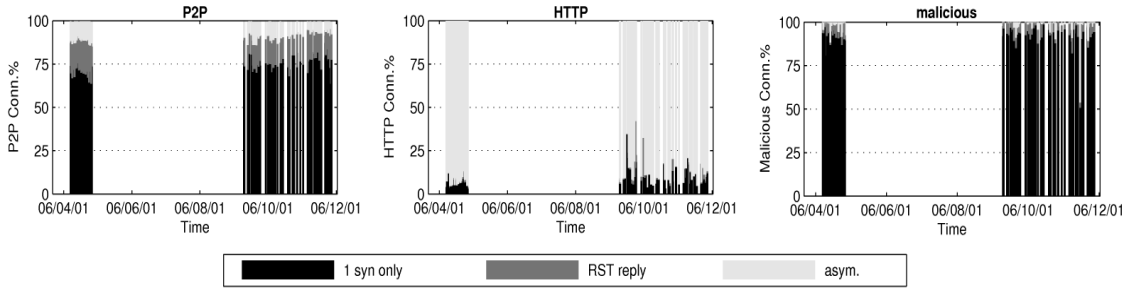


Fig. 3. Breakdown of non-established TCP connections

None of the traffic classes exhibits any significant diurnal pattern for non-established TCP connections. However, Fig.3 clearly highlights major differences between all three traffic classes. The already small fraction of non-established Web traffic (16.3% of all traffic) is mainly explained by asymmetrical traffic, and real unsuccessful connection attempts are very rare. Malicious traffic consists to a large degree of single SYN packet flows only. Single SYN flows are also dominating non-established P2P connections. While such connection attempts accounted for 71% in April, their fraction increase to 79% in November. This trend is also responsible for the increase of non-established P2P connections observed in Fig.2. Even if the high number of unsuccessful connection attempts within P2P traffic has been observed earlier [10], it is interesting to note that there is a clear trend in the fractions of one-SYN connections within P2P flows. The fraction increased by 23% (from 35% to 43%) within a period of 8 months.

Established Connections: Finally, established connections are broken down according to their termination behavior in Fig.4. Besides the proper closing approaches with one FIN in each direction or only one RST packet, as prescribed in the TCP standard, two unspecified termination behaviors have been observed. Connections closed by FIN, followed by an additional RST packet have been seen in direction of the initial SYN (typically the client) and the response (server). Finally, a number of connections were not closed during the measurement interval. The larger fraction of unclosed P2P connections is explained by the longer duration of P2P flows compared to Web traffic, as observed by Mori [7].

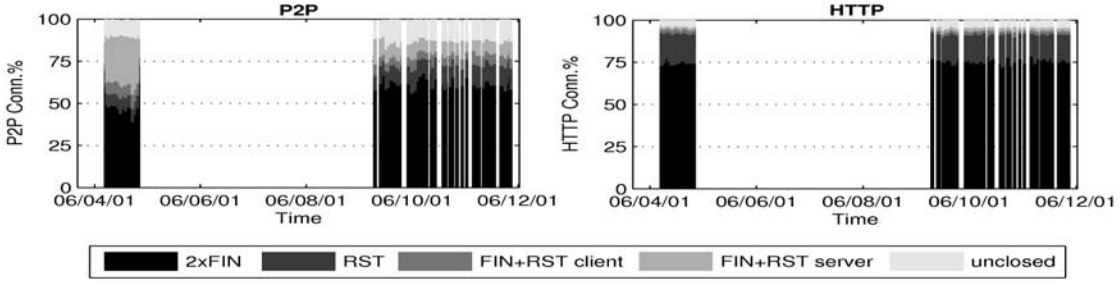


Fig. 4. Breakdown of established TCP connections

As for non-established connections, termination of Web connections neither shows significant trends nor diurnal patterns. HTTP connections are closed properly in 75% of all cases. Another 15% are closed by RST packets, mainly due to irregular web-server and browser implementations as noted by Arlitt [13]. FIN+RST behavior as well as unclosed connections (which corresponds to longer flows) are uncommon within Web traffic.

Even if there are no diurnal pattern observable, Fig. 4 indicates a significant change in termination behavior of P2P connections from spring to fall 2006. In April, only slightly less than half of the P2P connections have been closed properly with two FINs. As much as 20% of established P2P connections have been terminated with FIN plus an additional RST packet send by the server (or responding peer). A couple of popular hosts inside a student network have been identified as main source of this behavior. A commented text in the source code of a popular P2P client indicates that connections are closed with RST deliberately to avoid the TCP TIME_WAIT state in order to save CPU and memory overhead. In fall however, the fraction of FIN+RST terminations by the responder was reduced to around 8%, compensated by an increase in both valid TCP terminations, 2xFIN and single RST. Due to missing payload data, it was not possible to differentiate between different P2P software and version numbers. We suspect, that either the developers of the P2P application fixed this non-standard behavior in updated versions of the software, or the misbehaving P2P software lost popularity and was replaced by better behaving software by the users during 2006. However, the breakdown in Fig.4 shows that P2P traffic is mainly responsible for the large number of RST packets seen in todays networks.

5.2 Differences in Option Deployment

Finally, deployment of the most popular TCP options during connection established has been investigated for P2P and Web traffic (Table 1). For each of the four most popular TCP options, three different possibilities are distinguished: established - the option usage was successfully negotiated in SYN and SYN/ACK packets; neglected - the option usage was proposed in the SYN, but not included in the SYN/ACK; and none - the option was not seen in the connection.

	MSS	SACK	WS	TS
estab.	99.9%	91.0%	14.9%	8.8%
neglected	0.1%	6.5%	0.6%	1.0%
none	0.0%	2.5%	84.5%	90.2%

(a) TCP Options in P2P Conn.

	MSS	SACK	WS	TS
estab.	99.6%	65.7%	16.0%	13.4%
neglected	0.4%	27.9%	4.3%	4.3%
none	0.0%	6.4%	79.7%	82.3%

(b) TCP Options in HTTP Conn.

Table 1. Differences in TCP Option Deployment

Option usage turned out to be remarkably constant, with neither longitudinal nor diurnal trends. However, it is surprising to find such notable differences in option usage between traffic classes, considering that protocol stacks in the operating system, and not applications, decide about option usage. The MSS option is almost fully deployed, which agrees with the fact that the MSS option is set by default in all common operating systems. The SACK permitted option, in fact also a default option, is commonly proposed by initiating hosts, but is in 28% of the Web connections neglected. Interestingly, this fraction is significantly smaller in the case of P2P traffic, with only 6.5% neglecting SACK support.
While Linux hosts have the Window Scale (WS) and Timestamp (TS) options enabled by default, Windows XP does not actively use the options, but replies with WS and TS when receiving SYN packets with the particular option. This policy is well reflected by P2P connections, where WS and TS are rarely neglected, but either established or not used at all. HTTP connections do not really reflect this assumption, with 4.3% of WS and TS requests neglected by servers. However, WS and TS are established more often within Web traffic.
We suspect that the usage of WS and TS options within P2P traffic somewhat reflects the proportions of Linux (WS and TS enabled by default) and Windows systems (WS and TS disabled actively, but responding to request) on the links measured. The differences in option deployment for Web traffic however stem from a differing communication nature. While Web traffic represents classical client server communication, with one dedicated server involved, P2P represents a loose network of regular user workstations. Web-servers, as a central element, can thereby influence the behavior of larger numbers of connections. This suspicion is further confirmed by the fact that a majority of the HTTP connections neglecting usage of SACK are directed to less than 100 web-servers,

which consistently do not respond with SACK options. Such central elements do not exist in P2P overlay networks. Furthermore, web-servers are more likely to be customized or optimized due to their specific task, whereas user workstations usually keep default settings of the current operating system. Some active measurement samples taken in October 2007 proved that popular web-servers, like google, yahoo and thePirateBay, still neglect SACK, WS or TS options.

6 Summary and Conclusions

In order to study trends and differences within the main traffic classes on the Internet, aggregated backbone traffic has been collected during two campaigns in spring and fall 2006 [11]. The collected packet level data has then been summarized on flow level. The resulting connections have finally been classified into P2P, Web and malicious traffic, using a connection pattern classification method [9]. An analysis revealed that overall traffic volumes are increasing for both TCP and UDP traffic, with highest activities at evenings. On diurnal basis, P2P and HTTP traffic exhibit different peak times. P2P traffic was found to be clearly dominating with 90% of the transfer volumes, especially during evening and night times. In contrast, HTTP traffic has its main activities (9% of the data-volumes) during office hours. Similar diurnal patterns have been observed in terms of connection numbers, even if P2P connections are not as dominating as in the case of data volumes. This indicates that P2P connections typically carry more data than Web traffic. Malicious and attack traffic is responsible for a substantial part of all TCP connections and UDP flows, but plays a minor role in terms of data volumes since it typically consists of 1-packet flows only. It was interesting to observe that the fraction of malicious TCP and UDP flows remained constant in absolute numbers both on diurnal and longitudinal basis, even though traffic volumes generally increased. This shows that malicious traffic (e.g. scanning attacks) forms a constant background noise on the Internet.
In terms of connection signaling behavior, major differences between the three traffic classes have been highlighted. The number of unsuccessful P2P connection attempts, which already dominated the P2P connection breakdown in spring, was shown to have increased further until fall. We conclude, that the large fraction (43%) of 1-packet flows on one hand and the large average data amounts per P2P connection on the other hand indicate a pronounced 'elephants and mice phenomenon' (Pareto principle) [7] within P2P flow sizes. Regarding termination behavior, P2P connections exhibit a clear trend towards higher fractions of proper closings in fall. HTTP connections on the other hand appear to behave comparable well according to specification at all times.
Finally, also TCP option deployment was shown to differ significantly between P2P and Web traffic. While P2P traffic rather reflects an expected behavior considering the default setting in popular operating systems, HTTP shows artifacts of the traditional client server pattern, with some dedicated web-servers neglecting negotiation for certain TCP options. This is especially true for the SACK option. We conclude that even though SACK is deployed by almost all

P2P hosts and web-clients, a number of web-servers still neglect its usage. It is unclear to us, however, for which reasons web-server software or administrators would choose not to take advantage of certain TCP features, like SACK.
In the presented study, differences between traffic classes have been found in all aspects discussed, even if not always expected. The results provide researchers, developers and practitioners with novel, detailed knowledge about trends and influences of different traffic classes in current Internet traffic. The data analyzed was collected on a highly aggregated backbone link during a substantial time period, thus reflecting contemporary traffic behavior on one part of the Internet. Besides the general need of the networking and network security community to understand the nature of network traffic, information about behavior differences as seen 'in the wild' can be important when developing network applications, protocols or even network infrastructure. Furthermore, the results form valuable input for future simulation models.

References

[1] Moore, A.W., Papagiannaki, K.: Toward the Accurate Identification of Network Applications. Lecture Notes in Computer Science. (2005) 3431.
[2] Sen, S., Spatscheck, O., Wang, D.: Accurate, scalable in-network identification of p2p traffic using application signatures. WWW '04: Proceedings of the 13th Int. World Wide Web Conference, New York, NY, USA (2004)
[3] Crotti, M., Dusi, M., Gringoli, F., Salgarelli, L.: Traffic classification through simple statistical fingerprinting. Computer Communication Review **37**(1) (2007)
[4] Karagiannis, T., Broido, A., Faloutsos, M., Claffy, K.: Transport layer identification of p2p traffic. In: IMC '04: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, Taormina, Sicily, Italy (2004)
[5] Gerber, A., Houle, J., Nguyen, H., Roughan, M., Sen, S.: P2p the gorilla in the cable. In: National Cable and Telecommunications Association. (2003)
[6] Sen, S., Jia, W.: Analyzing peer-to-peer traffic across large networks. IEEE/ACM Transactions on Networking **12**(2) (2004)
[7] Mori, T., Uchida, M., Goto, S.: Flow analysis of internet traffic: World wide web versus peer-to-peer. Systems and Computers in Japan **36**(11) (2005)
[8] Perenyi, M., Trang Dinh, D., Gefferth, A., Molnar, S.: Identification and analysis of peer-to-peer traffic. Journal of Communications **1**(7) (2006)
[9] John, W., Tafvelin, S.: Heuristics to classifiy internet backbone traffic based on connection patterns. In: ICOIN '08: Proceedings of the 22nd International Conference on Information Networking, Busan, Korea (2008)
[10] Plissonneau, L., Costeux, J.L., Brown, P.: Analysis of peer-to-peer traffic on adsl. PAM '05: Proceedings of the 6th Passive and Active Network Measurement Workshop, Boston, MA, USA, Springer-Verlag (2005) 69–82
[11] John, W., Tafvelin, S.: SUNET OC 192 Traces (collection) Available: http://imdc.datcat.org/collection/1-04L9-9=SUNET+OC+192+Traces.
[12] John, W., Tafvelin, S.: Analysis of internet backbone traffic and header anomalies observed. In: IMC '07: Proceedings of the 7th ACM SIGCOMM conference on Internet measurement, San Diego, CA, USA (2007)
[13] Arlitt, M., Williamson, C.: An analysis of tcp reset behaviour on the internet. Computer Communication Review **35**(1) (2005)