Helsinki University of Technology
Department of Electrical and Communications Engineering
Networking laboratory

Markus Peuhkuri

Internet traffic measurements – aims, methodology, and discoveries

This thesis has been submitted for official examination for the degree of Licentiate of Technology in Espoo, Finland on May 28th, 2002.

Helsinki University of Technology
Department of Electrical and Communications Engineering
Networking Laboratory

Tekijä:	Markus Peuhkuri
Työn nimi:	Internet traffic measurements – aims, methodology, and discoveries
Päivämäärä:	28.5.2002
Sivumäärä:	102
Osasto:	Sähkö- ja tietoliikennetekniikan osasto
Professuuri:	S-38 Teletekniikka
Työn valvoja:	Professori Jorma Virtamo
Työn ohjaaja:	Professori Jorma Jormakka

Työssä on kehitetty menetelmä Internetin liikennemittaustallenteiden tiivistämiseksi hyödyntämällä IP-verkkoliikentessä samaan yhteyteen kuuluvien pakettien keskinäistä samankaltaisuutta. Tiivistysmenetelmän yhteydessä voidaan toteuttaa myös osoitteiden anonymisointi kuitenkin säilyttäen verkkoliikenteen topologiainformaatio ja kohteiden yksikäsitteisyys.

Verkkoliikennettä kerättiin 18 kuukauden ajalta ja mittausaineistoa analysoitiin tarkoituksena havainnoida verkkoliikenteen perusominaisuuksia. Aluksi eri sovellusprotokollatyyppien liikenteellisiä ominaisuuksia analysoitiin määrittelyiden ja esimerkkien pohjalta.

Vuo, joka Internet-liikenteessä muodostuu pakettijonosta, voidaan määritellä eri kriteereillä, joista tärkeimmät ovat määrittelyn yksityiskohtaisuus ja pakettien välisten aikojen maksimipituus. Voiden määrää tutkittiin näiden kahden kriteerin perusteella, jolloin havaittiin eräiden sovellusprotokollien olevan hyvin herkkiä raja-arvojen valinnalle.

Käyttäjän kärsimättömyys voi aiheuttaa hukkaliikennettä verkkoon. Tärkeimmiksi HTTP-protokollan kärsimättömyystekijöiksi osoittautuivat 10 sekunnin siirtoajan ylittyminen ja alhainen siirtonopeus. Kaistanleveys päätelaitteiden välillä osoittautui hyvin vaihelevaksi lyhyeilläkin aika-asteikoilla, joten yhteyden varmentaminen testiliikenteellä ei aina takaa yhteyden laatua.

Avainsanat: liikennemittaukset, QoS, käyttäjän kärsimättömyys

Author:	Markus Peuhkuri
Title of the thesis:	Internet traffic measurements – aims, methodology, and discoveries
Date:	2002-05-28
Number of pages:	102
Faculty:	Department of Electrical and Communications Engineering
Professuuri:	S-38 Telecommunications Technology
Supervisor:	Professor Jorma Virtamo
Instructor:	Professor Jorma Jormakka

A method to compress Internet traffic packet traces, utilising similarities between packets within a connection, was developed. Along with compression, packet address fields can be anonymised while maintaining topology information and uniqueness of end hosts.

Internet traffic traces were collected over an 18 month period. This data has been analysed in order to identify basic properties of network traffic. Traffic properties of different application protocols were studied based on specifications and examples.

A flow, built up from a train of IP packets, can be defined by various criteria of which the most important are address granularity and packet interarrival time. Counts of flows were studied based on these criteria. It was found that some of application protocols were very sensive to the threshold selection.

User impatience may result in useless traffic being carried in the network. The most important factors for impatience turned out to be a transfer time exceeding 10 seconds and a low transfer rate. Available bandwidth between hosts was found variable even over short time scales. It was concluded that using test traffic probes does not guarantee quality of connection.

Keywords: traffic measuremets, QoS, user impatience

Preface

This work has been made for major part in the Mi^{2}tta project funded by Academy of Finland.

First of all, I like to thank my supervisor professor Jorma Virtamo and my instructor professor Jorma Jormakka for guidance during this work. I received enough pressure to finish this work earlier than "by end of next term" and before the title have been "the universe with three examples".

My colleagues, Lic.Tech Marko Luoma and Lic.Tech Mika Ilvesmäki, as members of "3M", have provided input in bilateral exchange of ideas, even bad ones.

Other members of the Networking laboratory coffee room gang have provided nice breaks from normal routines – "you know who you are", especially to mention Kimmo Pitkäniemi and Arja Hänninen who have helped with administration.

I thank you, my loved ones: Katri, Saara, and Iiro, for giving me a bunch of joy for my life in our nest.

Contents

Acronyms and terms

2s-MMPP	Two-stage Markov Modulated Poisson Process
ACK	In TCP: acknowledgement flag or segment acknowledgementing data
AS	Autonomous System
ASCII	American Standard Code for Information Interchange, the most widely used 7-bit character set which covers letters “a” to “z”. Specified in ANSI X3.4 and in international counterpart ISO 646.
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol, external routing protocol used in Internet
CAC	Call Admission Control
CBR	Constant Bit Rate
CDR	Call Detail Record
CDF	Cumulative Distribution Function
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSQ	Squared Coefficient of variation
DDoS	Distributed Denial of Service
DNS	Domain Name System Internet name service
DoS	Denial of Service
DQDB	Distributed Queueing Dual Bus, a metropolitan network technology.
DS	Differentiated Services
ECB	Electronic Codebook, a way to use block chipper so that each block is independent – does not depend on previous block(s) – same data map always to same encrypted value.
ECM	Explicit Congestion Notification, experimental mechanism that network can use to signal end systems about congestion without dropping packets [RFB01]

EOP	End of Option List, TCP option indicating the end of option list [Pos81c, p. 18].
FBM	Fractional Brownian Motion
FGN	Fractional Gaussian Noise
FIN	A flag indicating end (finish) of TCP connection
FR	Frame Relay
FTP	File Transfer Protocol
GSM	Global System for Mobile communications
GPS	Global Positioning System
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol over TLS
ICMP	Internet Control Message Protocol
IDC	Index of Dispersion of Counts
IEEE	Institute of Electrical and Electronics Engineers, a non-profit, technical professional association with consensus-based standards activities.
IETF	Internet Engineering Task Force
IMAP	Internet Message Access Protocol
IMAPS	Internet Message Access Protocol over TLS
IN	Intelligent Network
IP	Internet Protocol
IPPM	Internet Protocol Performance Metrics, an IETF workgroup
IPSec	IP Security protocol providing authentication and encryption at IP level [TDG98]
IPv4	IP version 4, the current IP version
IPv6	IP version 6, the new version of IP
ISDN	Integrated Services Digital Network
ISP	Internet Service Provider
ITU-T	International Telecommunication Union, Telecommunications sector
kibi	kilobinary, $2^{10} = 1024$. A binary prefix standardised in IEC 60027-2. `http://physics.nist.gov/cuu/Units/binary.html`
LAN	Local Area Network

LDAP	Lightweight Directory Access Protocol
LRD	Long-range Dependence
MAC	Media Access Control, a sublayer of OSI layer 2 in local area network standards defining addressing among other things.
MAN	Metropolitan Area Network
MD5	Message Digest 5, an algorithm to calculate cryptographically safe message digest [Riv92].
MIB	Management Information Base, a set of SNMP manageable variables.
MMPP	Markov Modulated Poisson Process
MPEG	Moving Picture Expert Group
MPLS	Multiprotocol Label Switching
MTA	Mail Transfer Agent
MUA	Mail User Agent
NIMI	National Internet Measurement Infrastructure
NNTP	Network News Transport Protocol
NOP	No-Operation TCP option that can be used to align subsequent options at word boundary [Pos81c, p. 18].
NPD	Network Probe Daemon
ns-2	Network Simulator version 2, network simulator used among IETF workgroups, available at `http://www-mash.CS.Berkeley.EDU/ns/`.
NSFNET	National Science Foundation Network
NTP	Network Time Protocol [Mil92]
OC3MON	OC3 monitor
OSI	Open System Interconnect
OSPF	Open Shortest Path First
PC	Personal Computer
PDF	Probability Density Function
PDU	Protocol Data Unit
PMR	Peak-to-Mean Ratio
PMTU	Path Maximum Transfer Unit Maximum size transfer unit that can travel network end-to-end without being fragmented.

POP	Point Of Precedence
PPS	Pulse Per Second, a reference clock output, which emits one pulse once a second.
PRNG	Pseudo Random Number Generator
PSTN	Public Switched Telephone Network
QNA	Queueing Network Analyser
QoS	Quality of Service
RST	A flag indicating reset of TCP connection
RTO	Retransmission Time-out
RTP	Real-Time Control Protocol
RTP	Real-Time Protocol
RTT	Round Trip Time
SACK	Selective Acknowledgement, a method to inform sender about received non-continuous segments. Also a TCP option to implement this [MMFR96].
SLA	Service Level Agreement
SMTP	Simple Mail Transport Protocol
SNMP	Simple Network Management Protocol
SPI	IPSEC Security Parameter Index
SPS	Standard Positioning Service
SRV	"Location of Service" record
SSH	Secure Shell
SYN	A flag indicating start of TCP connection by synchronising sequence numbers.
TCP	Transmission Control Protocol
TLS	Transport Layer Security [DA99]
TReno	Traceroute RENO
TTL	Time To Live
UDP	User Datagram Protocol
URI	Uniform Resource Identifier
URL	Uniform Resource Locator

UNIX	UNIX
vBNS	Very high performance Backbone Network Service, part of the Internet2 initiative.
VHF	Very High Frequency
VoIP	Voice over IP
WG	Work Group
WWW	World Wide Web
X11	X11 Window System

Chapter 1

Introduction

1.1 Needs for traffic measurements

Traffic measurements are performed for a variety of reasons. These include traffic and network characterisation, network monitoring and network control. The last two are closely related to each other and differ by time scales on which they operate.

1.1.1 Traffic characterisation

One of basic building blocks for network analysis is the knowledge of workload or traffic volume demanded by network users. It should not be assumed, however, that the traffic volume is a fixed quantity. There are multiple factors that either increase or decrease the capacity demand. A well functioning network attracts more users and new ways to utilise the network thus increasing traffic demand, while an inappropriate pricing or an unreliable network could reduce demand, especially if there are alternative ways to communicate. A high rate of failed connections or discarded packets may result in an increase of traffic demand.

Traffic characterisation has been utilised in the telephone network since the beginning of the 20th century, when Erlang established foundation for modern traffic theory [Kiv94]. In the early years, traffic data was collected by hand while collecting information for charging. In the 1920s, traffic was estimated by sampling subscriber and trunk lines randomly and traffic was estimated based on this. The quality of connection and blocking rate was also recorded. In the 1950s and 1960s automatic sample based measurement devices were developed [Rah01].

For a network operator, proper network dimensioning is an essential task. Too much capacity generates unnecessary expenses, too little causes customer dissatisfaction. A statistical analysis for traffic pattern identification and in particular locating peak patterns with their variations enables one to find daily, weekly, and seasonal variations.

Traffic distribution is a second factor that has an impact on network dimensioning and network topology. Capacity has to be in a right location to carry offered traffic. Traffic distribution as well as traffic volume may vary by time: during work hours there is much traffic demand for business-to-business communications while at night there is more leisure-related traffic.

Source characterisation is mainly related to network research activities. The objective is to find appropriate models for analytical and simulation studies. This is further discussed in Section 2.3.3.1. A related area is network characterisation. The communication networks are complex and constantly changing systems. There are both practical and political reasons, such as trade secrets, why operators do not provide information about their networks. Methods that are developed to characterise networks are reviewed in Section 2.2.4.3.

Introducing QoS methods to the Internet [BCS94, BBC+98] brings yet another dimension to traffic volume estimation: there will be different mix of service classes at different times and in different locations. Each service class may receive different treatment and a different route in the network.

1.1.2 Network monitoring

Network monitoring is an essential part of network operation and maintenance. While monitoring may utilise same tools and data as traffic characterisation, its objectives are shorter term. Network state identification and fault detection are the two most important objectives. Other objectives include quality monitoring, intrusion detection, policy evaluation, and agreement verification. These are explicated in detail in Section 2.2.5.

1.1.3 Traffic control

Measurements can be used in real-time traffic control by optimising network routing and path selection according to traffic load in different links. It is also possible to use real-time measurement data to support measurement-based admission con-

trol. These mechanisms are outside the scope of this work.

1.2 Terminology

There are some terms that may have a different meaning in different contexts and when discussing different technologies. Following are definitions used in this work.

Throughput is the maximum sustainable rate at which a network or network element can deliver information considering present network conditions and QoS objectives.

Goodput is the maximum sustainable rate at that an *application* can transmit useful information under present network conditions. Unnecessary retransmissions and data which arrive late are not included in this figure. This is also referred to as *application throughput*.

Traffic volume is the amount of data delivered to the network to be transmitted to the destination.

Bandwidth is the amount of data delivered by a network at the location of the measurement. Note that this may differ from the goodput as it includes also "useless" data for an application, such as unnecessary retransmissions.

1.3 Measurement classification

Each measurement has two factors that define the measurement population: location and granularity. The location defines which part of a network is measured and what traffic is included in the measurements. How the location is selected has a significant effect on the usefulness of the measurement. The measurement location should be representative for traffic or network to be measured. For example, if the network end-to-end performance is to be measured, the measurement system should be located at a "typical" part of an access network, not at the edge of the core network. Similarly, if one is interested about the core network performance, measurement locations should be at the edge of core network, not in access networks.

If measurement data are collected by the network elements themselves, some care should be taken considering the load caused to the operational network

elements. Extensive statistics collection may result in performance degradation. The network elements are optimised for data delivery. Secondary functions such as measurements and diagnostic replies[1] operate at lower priority compared to data forwarding functions. In the event of high load, reply requests are ignored or delayed which may skew results. External measurement equipment does not have these disadvantages – except for using network capacity to transfer measurement data – but they introduce additional cost.

1.3.1 Time scales of measurements

Network measurements can be performed continuously or sample-based. Some measurements, such as counting bits and packets, can be performed continuously even on high link speeds. Measurements that require maintaining the state of individual flows or their payloads, like HTTP transaction analysis [Fel98], may not be worth the resources needed for continuous measurements in the core network.

Some measurements are sample-based by nature, for example testing packet loss and delay by sending test traffic [AKZ99a, AKZ99b]. Another group of measurements does have utility only as continuous measurements: such measurements include identifying busy period or maximum network load.

1.3.2 Flow-based measurements

Certain measurements in a telephone network are carried out on the call level using call detail records (CDR). For each call the system records

- start time (`ST` on Figure 1.1),
- duration (`CT`)[2],
- caller number (A subscriber, `AS`),
- callee number (B subscriber, `BS`), and
- reason for tear down (`CC`).

[1] Such as sending ICMP messages.

[2] There are several start times and durations recorded for different phases of telephone call

```
EXCHANGE = 039082  NAME = TST  DATE = 2000-08-18 11:18:53.35

NGC: -                                  CRI: -
 SE: -                                   SS: 0000
 AS: 9055003                             BS: 50100
FSU: -                                  FAN: -
CFC: -
 ST: 2000-08-18 10:31:52.74    ET: 2000-08-18 10:32:06.74   CH: 0000
 CT: 0          CP: 0          RP: 0          AT: 01 AC: 00 AF: 0000
AFC: -    -    -    -         BFL: -         BFC: -    -    -    -
AFT: -         LBS: -                        BFT: -
 CC: 00000000   DT: 00200001   CI: 103102F4   TI: 00 BE: 10 TS: 0000
 UC: 00000000   UT: 00000000   PB:
 TB: -                                 US: -          AU: -
```

Figure 1.1: An example of call record ("ticket").

These data "tickets"[3] can be readily downloaded from modern computer controlled telephone exchange and are used for ticket-based customer charging. An example of telephone exchange call record is shown in Figure 1.1, which includes only some of possible timestamps [Nok]. The CDRs can be collected only for every chargeable call (toll-ticket), for every call (all-ticket) or by sampling basis for traffic measurement purposes [E.497]. In the ITU-T E-series, there are several recommendations for various service quality factors that can be calculated based on data in "tickets" [E.497, E.496]. One of the interesting factors is call setup time [E.492].

In IP traffic the equivalent is *flow-based* measurements. Packets are grouped into flows by their source and destination IP addresses, protocol and source and destination port numbers[4]. For each flow a set of details is collected. These include the starting timestamp of flow, flow duration, packet count, byte count, etc.

As there are a huge number of flows in the Internet – one web page retrieval may result to several TCP flows – flow measurements produce a lot of data. Maintaining state information of concurrent flows in the core network may be a difficult problem [Wan01]. Flow measurements are deployed in access networks mainly. Number of concurrent flows is limited and flow data volume is not too large. Flow data can be used for accounting purposes.

[3]The name "ticket" originates from time of manual telephone exchanges. The switchboard operator kept record about each call. The record was called a "ticket".

[4]If the protocol in question has those. The UDP and TCP protocols presently contributing most of the traffic in the Internet, do have port numbers.

One can use also more coarse-grained criteria for a flow, thus reducing the total number of flows. See Section 5.1.1 for different possible flow granularity. If some routing-related information such as MPLS paths is used as definition for a flow, a network equipment may have statistics collection readily available.

1.3.3 Network element based measurements

A network consists of network nodes; each node consists of one or more network interfaces, which are connected via links to other interfaces in other nodes. Network nodes maintain counters for each interface for sent and received data. These counters include number of packets, byte count, and number of packets discarded or with error. Node-wide counters for forwarded packets are available. This data can be retrieved for example by SNMP with a relatively low overhead.

This measurement is monitoring, i.e. passive and local. It does not give an overall view of the network status or direct indication of the end-to-end performance. However, by combining statistics from multiple network elements, a better view can be achieved.

1.3.4 Node to node measurements

Performance monitoring is probably the most popular application of node-to-node measurements. A typical use of active measurement is to send test traffic, e.g. such as specified in the IPPM framework [PAMM98]. Measurements can be carried between end systems to measure end-to-end performance or between some intermediate systems to monitor some part of the network.

Traffic used for measurement can be either real application traffic or test traffic. As noted above, ICMP messages may have a different treatment in both end systems and in routers. This may result in a view that is either too optimistic – in case the system is over-loaded but can still provide in-kernel responses to ICMP messages – or too pessimistic – in case ICMP processing is delayed because of high load.

1.4 What is measured

There are multiple quantities to be measured from the network. Depending on whether one is interested in offered traffic, network performance, network characteristics, traffic characteristics, node performance, or other type of measurement, a different set of measurement entities is selected.

1.4.1 Quantitative measures

Traffic volume is one of the most interesting measurement entities. Offered traffic dictates how a network should be dimensioned. Traffic volume can be characterised by the mean volume and the variance of the volume or peak period volume. Unit can be bits, bytes, or packets, depending on the network and equipment. Total volume can be further divided by flow (or macro flow, see Section 5.1.1), traffic type, service class, or network element where it is measured.

Measured traffic volume should be analysed in a cautious manner. The measured traffic volume may differ from the offered traffic because of congestion or network failures. User or application behaviour may change because of unsuccessful attempts; whether this leads to recalls or abandonment should be carefully explored.

The difference between the traffic volume and the network capacity is the *available bandwidth*. This can be utilised for load balancing or measurement based admission control. Available bandwidth is always a link-local figure and in general is different even in neighbouring links.

Closely related to the available bandwidth is the *call holding time*. Even if there may not be actual "calls" in the network in question, it is an indication how long a customer or a connection uses network resources. This has an effect on network dynamics: how fast resources will be freed if no new customers are accepted and how long a customer will keep resources reserved.

1.4.2 Qualitative measures

From network users' viewpoint probably the most interesting entity is network *throughput*. This is the amount of useful transfers and in general is less than traffic volume because of lost, retransmitted, errored, or misdelivered packets. It should be taken into account at which protocol level the throughput is measured because

of protocol overheads.

Packet *delay* and its *variation* are other components for user satisfaction. Changes in network load can first be seen in changes of delay. In an unloaded network packet delay is the sum of packet processing, serialisation, and propagation delays. When network load increases, packet experiences queueing delays, which increases total delay.

There are several methods to measure delay variation. In [Y.102, Appendix II] four methods are defined. The first two methods give the variation for an individual packet as the difference to either the delay of the first packet or to the average delay of the population. The third alternative is to define a delay interval in advance and count the proportion of packets that fall outside this interval. The last alternative is to measure the distance between two quantiles (like 0.95 and 0.5).

For example, if a set of packets are transmitted over a network and the first packet receives a delay of 388 ms, and the average delay is 374 ms, then the variation for a packet with delay of 402 ms would be 14 ms by the first method and 28 ms by the second method. If the pre-defined delay bounds are 330–380 ms then the delay variation based on the third method would be 15 % to whole population, because 3 % packets received shorter delay than 330 ms and 12 % longer than 380 ms. The delay variation by the last method would be 47 ms as the median delay is 348 ms and the 95 % quantile is 395 ms.

There are several sources for *packet loss*. While transmission or protocol errors result in packet losses that are not caused by excess traffic, they do give indication of network quality. Packets may be deliberately lost because of policing on network edge. This is not a sign of network performance problems but just enforcing of the traffic agreement. Network congestion and packet loss because of buffer overflow or active queue management methods can be a sign of insufficient resources.

It depends on the application and transport protocol what level of packet losses is acceptable. During network failures and routing updating there may be periods of complete packet loss. If one simply includes these periods in long-term statistics, the results may not be informative. It is better to introduce new statistics such as burst loss or severe loss ratio counts or durations. This makes it possible to differentiate between “normal operations quality” and “outages”.

Yet another measure that may not be directly observable from network traffic or to network users is resource usage. However, it is important for a network operator

Table 1.1: Who cares about measurements [CM97].

	Goal	Measure
ISP	• capacity planning • operations • value add services (e.g. customer reports) • usage-based billing	• bandwidth utilisation • packets per second • round trip time (RTT) • RTT variance • packet loss • reachability • circuit performance • routing diagnosis
Users	• monitor performance • plan upgrades • negotiate service contracts • set user expectations • optimise content delivery • usage policing	• bandwidth availability • response time • packet loss • reachability • connection rates • service qualities • host performance
Vendors	• improve design and configuration of equipment • implement real-time debugging and diagnosis of deployed hardware	• trace samples • log analysis

to know how much resources there are to accommodate a possible increase in traffic. Critical resources include router processor utilisation, buffer occupancy, forwarding table size, and link utilisation.

1.5 Network measurement motivation

Some motivation for traffic measurements is given in Table 1.1. The paper [CM97] discusses the current state of Internet traffic measurements and describes goals of the CAIDA project: creating a set of Internet performance metrics with IETF IPPM, creating an environment to share data confidentially or in desensitised form and fostering the development of advanced networking technologies such as multicast, caching and QoS. Also much emphasis is put on visualising measured data.

In addition to normal utilisation statistics obtained via SNMP MIB variables, both long-term aggregated statistics and short-term per flow statistics provide essential insights relating to [Mea]:

- network provisioning,
- peering arrangements,
- per-customer accounting and SLA verification,
- per-per accounting (traffic balance of trade),
- performance management,
- tracking topology and routing changes,
- tracing DoS attacks,
- ATM/cell/circuit level errors and other troubleshooting,
- connectivity complexity and vulnerability,
- TCP flow dynamics, and
- routing table and address space efficiency.

1.5.1 Operator requirements for measurements

A network operator has different needs for network measurements than a researcher. A long lifetime for network investments and continuity of operations are important factors for an operator. The measurement system must be able to support traffic engineering and different platforms and protocols. There is need for a well-defined measurement standard for interoperability that is more than just a common protocol to exchange measurement data. Any inconsistencies in statistical definitions, protocol levels, or data collection should be avoided.

A measurement system must be able to scale with the size and speed of the network. The information should be aggregated as much as possible without losing essential details. The measurement activity must not load network elements to such a degree that the primary function, delivering information, degrades.

1.5.1.1 Network operator time scales

Three different time scales can be identified for a network operator to react on network measurements. The demand for measurements depends on the timescale it is used for. The longest time scale, order of months, is for network planning. This includes network extension or introducing new technologies to meet future needs for capacity and reliability.

Capacity management reacts on the time scale of hours or days. Using existing capacity and equipments the network is reconfigured to optimise utilisation. Real-time network control can be manual or automatic and works in minutes or shorter time scales. Its objective is to apply short-term corrections to network configuration in event of congestion or failure. Capacity management reviews these corrections at a later time .

1.5.1.2 Relationship of performance criteria

ITU-T has defined a 3x3 performance framework in [I.393]. There are three protocol-independent protocol functions: access, user information transfer, and disengagement. All of these functions are considered according to three performance criteria: speed, accuracy, and dependability. The performance criteria are protocol-dependent as criteria for circuit-switched N-ISDN are not applicable to a packet-switched protocol such as IP. Together these criteria define a set of performance requirements: if a service fulfils these requirements, the service is declared to be "available", otherwise the service is "unavailable". In [Y.100] relationships between ISDN, IP, and GII performance requirements are described. It also includes a list with short descriptions about each relevant ITU-T recommendation.

1.6 About this work

This work consists of four different parts. The first part gives background of network measurements by reviewing literature. The second part introduces the developed measurement system with data compression and capability to remove sensitive information from network traces. In the third part a set of discoveries from measured data are represented. The last part includes discussion of the role of network measurements and the final conclusions.

Chapter 2 is a survey of network measurements with the focus on Internet mea-

surements. Firstly active measurements are discussed starting from simple test traffic and ending with large-scale measurement infrastructure projects. Passive measurements include discussion of different statistics, which can be calculated based on measurement data. Network measurements from a fast and live network have certain problems, which are discussed in Section 2.5. Finally, a set of tools used for network measurements is briefly reviewed.

Network measurements, which are later analysed, are described in Chapter 3. A method to efficiently compress packet traces and to remove sensitive information while retaining as much information as possible from the traces is presented [Peu01].

The first analysis deals with identifying different network traffic classes in Chapter 4. Different protocols are studied based on their specifications and common implementations. Based on that, characteristics of network traffic are estimated.

A flow, which can be defined by multiple criteria, is another fundamental unit of data in addition to IP packet in the Internet. Different flow definitions are discussed in Chapter 5. Measured data is analysed and properties of flows are analysed.

User impatience is one of factors affecting network utilisation and user satisfaction. A method to identify user impatience is introduced in Chapter 6 and the results are discussed.

Stability of network throughput is one interesting question: if one measures network throughput at certain time, will the throughput be the same after some time period? This is studied in Chapter 7.

The need of network measurements is discussed in Chapter 8 considering network operators view and user's view. The objectives do differ depending on which side of an access router one acts. Finally, this work is concluded in Chapter 9.

Chapter 2

Survey of IP network measurements

In this chapter first the concept of measurements is studied, both for active and passive measurements (Sections 2.2 and 2.3) including a review of literature and efforts in the area of network measurements. Secondly, a representive set of measurement tools are described in Section 2.4. Additionally, technical and legal problems in measurements are discussed in Section 2.5.

2.1 Internet Measurements

One of the first Internet measurement papers is [KN74]. As the Arpanet was designed to be an experimental network, there were extensive facilities for data collection [CPB93]. Several characteristics were investigated in [KN74, CPB93]:

1. Message size and packet size distribution. While "message" is not applicable in the current Internet [CPB93] because of diverse link-layer technologies, one of additional metrics could be length of flow: for example the length of one web document.
2. Delay statistics.
3. Mean traffic-weighted path length.
4. Incest (traffic destined to same site: not applicable with the current architecture).
5. Most popular sites and links.
6. Favouritism (a site communicates much with a small number of sites).

7. Link utilisation.
8. Error rates.
9. Attributing long-term growth to domains and protocols.
10. Trend of average packet size over different time scales.
11. International distribution of traffic.

2.1.1 Problems with multi-provider environment

As NSFNET backbone was discontinued by April 1995 extensive statistics collection was discontinued also. As the network was operated by different commercial service providers, the statistics collection became much more difficult, interfered by cost-benefit tradeoffs [BC95].

In current equipment the main emphasis is in fast and reliable packet delivery and collection of metrics for traffic engineering has not developed at the same rate as equipment forwarding and routing capacities [Mea].

The network metrics can be classified into at least four categories [Lam95]:

Utilisation metrics: packet and byte counts, peak metrics, protocol, and application distribution.

Performance metrics: round-trip time (at different layers) and packet drop count. Other suggested in [Lam95] were collision count at bus networks and ICMP source quench message counts. The use of ICMP source quench is not recommended [BE95, p. 57] and the use of early congestion notification (ECM) mechanism by marking packets is being introduced [RFB01].

Availability metrics: long-term line, route or application availability.

Stability metrics: short-term fluctuations that degrade performance such as line status transitions, route changes, next hop stability and short term ICMP anomalous behaviour.

The measurements can be divided into two categories: active measurements (Section 2.2) and passive measurements (Section 2.3). The former group involves sending data, either real application data or measurement-only data, and measuring

time either at both ends or the response at the sending end only. A measurement can cause excess load on the network and the results can be biased towards worse results than the actual service received. It is also possible that the test traffic receives better quality of service and results are better than the actual performance.

Passive measurements do not add traffic to the network, expect for transfer of results, which can be done also off-line. On the other hand, one cannot measure connections if there is no traffic. Privacy issues present another problem with passive measurements if the measurements are done on an operational network. Some measurement efforts may combine both passive and active measurements.

2.2 Active measurements

In an active measurement one sends data to the network, measures the response and checks the result for important factors (latency, jitter, throughput, packet loss). Active measurements may cause a disturbance to the network by introducing excess traffic.

The measurement can take place at both ends – when specialised software or hardware is needed at both ends – or only at one end. In the latter case, there may be no need for special equipment at the remote end. In the case one is interested in end-to-end application performance, this is satisfactory. The application performance may be affected by several other non-network related factors such as server load or latency in back end systems [Nie99] that will increase latency. One must make sure that one is measuring the correct metrics.

Each measurement packet tells only how much it was delayed in transit or if it was lost. The measurement of a single packet does not tell very much about conditions in the network. Several packets are needed to get accurate information about the network state at a certain moment.

Active measurements can be divided into three categories based on how they give information of the network structure and the state of the network.

End-to-end measurements tell only what the end-to-end characteristics of the network are (Figure 2.1). This is the only measure that is of interest to an end user: he is interested in the total quality of an application. Typical measurements are availability, delay, and throughput measurements. The

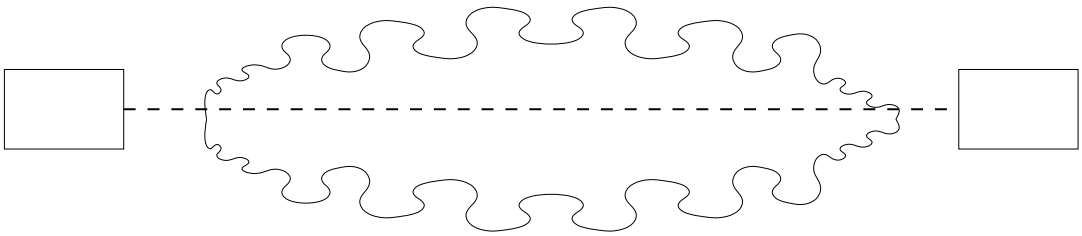


Figure 2.1: End-to-end measurements.

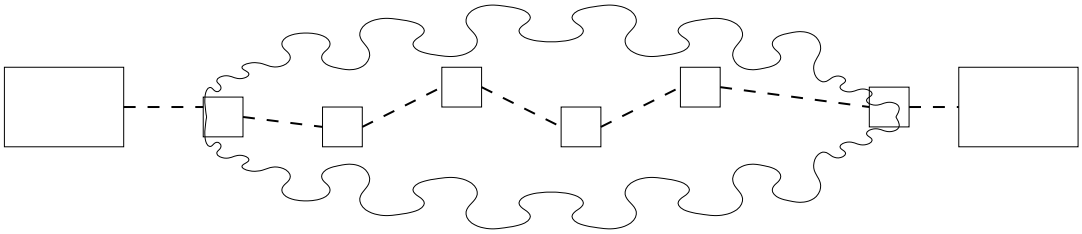


Figure 2.2: Hop-by-hop measurements.

majority of practical measurements are in this category.

Hop-by-hop measurements try to tell the state of each hop on path of packet (Figure 2.2). A typical tool, traceroute, gives the delay to each node on the path. From those results, one can find out if there is some problematic link.

Link-by-link measurements differ from hop-by-hop measurements in the sense that they try to characterise also the links between the nodes (Figure 2.3). This is quite difficult a task, but there are some tools available as the ones described in [Jac97, Dow99, CC96b].

Most of the tests for end-to-end availability and delay are based on the ICMP Echo Requests and Responses. The ICMP protocol is designed to take care of error reporting, configuration, information, and diagnostic tasks for IP [Pos81a, Pos81b]. As every host and router must implement ICMP [BE89], it is readily available.

Another popular use is to send UDP or TCP packets [Pos80, Pos81c] with increasing TTL field values. This, in principle, makes it possible to find all nodes

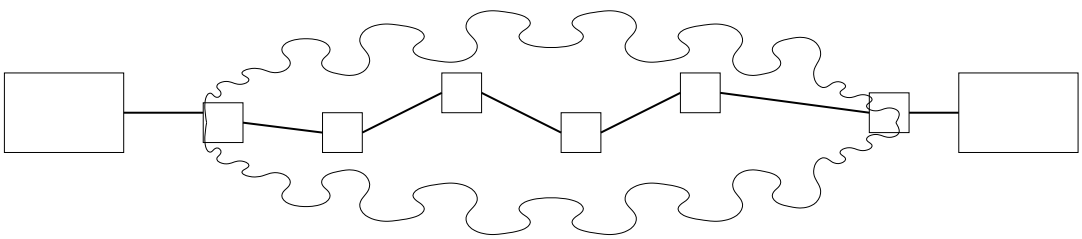


Figure 2.3: Link-by-link measurements.

on the path when routers reply with ICMP Time Exceeded messages. When TTL value is increased by one, the next router or end system is identified.

2.2.1 UDP and TCP simple services

Traditionally, UNIX systems have had so called "simple services" which include "echo", "chargen" and "discard". The echo service echoes all the data sent to it back to the sender, the chargen outputs lines of printable ASCII characters so that each character will be in every position in turn. It is suitable for checking line printer output and also for some data dependent transmission errors. The discard service works as a sink.

Leaving those services enabled is nowadays not recommended as they make possible some denial-of-service attacks [LP99]. To protect the system, most of modern implementations also have some form of rate limit that limits the usability for performance testing.

As those simple services are less and less available, the only way is then to use the actual services, like HTTP server. This, of course, requires that the end system has the server installed. This limits the test applicability to arbitrary hosts.

2.2.2 Limitations of non-application protocol tests

There are, however, a few points to keep in mind when using a protocol other than that of the application whose performance one is interested in.

1. The network may provide different level of service for different protocols. For example the UDP traffic could have lower priority than ICMP or TCP. Also port numbers used with TCP and UDP can have different priorities. For example, the web traffic (port 80) may have higher priority than network news (port 119). This may affect performance tests.

2. Some types of traffic may be administratively blocked. In this case a false negative result may result in connectivity tests. Also some types of traffic may have some kind rate limit: for ICMP messages there is a rate limit in the current versions of Linux and Solaris [Jac97].

3. Application traffic profile may be different from test traffic and some consequences of congestion or overload are not detected by test traffic [BG98,

CJ99]. The application fidelity may not be easily derived from simple loss and delay figures.

4. The vast majority active measurements send periodic streams of packets while the application traffic is typically very bursty. At times of high load, when there can be QoS problems and large amount of application traffic is carried, the proportion of test packets is low. This results in underestimating the times of low QoS [KH02].

To explain the mechanism in 3. consider a MPEG video transmitted over an IP network. One MPEG frame may be fragmented into several IP datagrams. Based on the MPEG-1 frame size distribution of the "terminator"-trace described in [Ros95], 31 % of frames needed more than one IP datagram when the RFC 2250 encapsulation and maximum datagram size of 1,500 bytes were used [HFGC98]. While more than 85 % of B-frames fit into one datagram, every I-frame needed at least two datagrams and 25 % needed four or more. In a test where a tele-immersion application was simulated over vBNS network, there were several periods of severe frame loss (more than 50 %) while the packet loss was less than 5 % [CJ99].

2.2.3 Application based performance measurements

A simple way to measure received quality is to measure the actual applications. A typical example is to periodically retrieve certain web pages and to measure the time needed for download. This is very simple to accomplish if the end user is interested checking his own connection relative to the services he needs. It is more complicated if the service provider (like a web server owner) wants to know if his customers receive documents quickly. This would include setting up client systems on various locations.

The network provider and the customer usually agree on some level of service the customer is receiving. This is called the service level agreement (SLA) and can be either implicit or explicit. Both the customer and network operator are interested in how well one complies with that agreement and how the network is operated. This can be done either actively transmitting test traffic or passively observing the network and measuring traffic generated by users. The first approach has problems as it increases load in the network and also there are problems in selecting representative test traffic targets. The latter approach does not have these problems as users "automatically" select "important sites" as they use the

network.

Paper [Asa98] discusses how to identify representative measurements to avoid false alarms and also the selection of the baseline. They use logs from a HTTP proxy as measurements, as throughput can be calculated from logged entries. Active throughput measurements were used to validate results drawn by analysing logs. It was found that the active and passive results disagreed for about 10 % of time: most of that included time periods when passive measurement had low reliability.

There are several tools to generate traffic and measure throughput. The major shortcoming with those tools is that they do not resemble real applications. A general-purpose traffic generator-analyser (KITS) is described in [MIIA99]. It uses the RTP protocol [GSC+96] with auxiliary fields to improve both resolution and protection for sequence number wrap-around. The KITS was used to generate background phone-type traffic in subjective phone quality tests. Characteristics measured by KITS and subjective ratings were compared. Accurate synchronisation with network time protocol (NTP) was found problematic and the use of GPS was suggested to utilise full potential of the KITS resolution. Another similar tool is NetSpec [LFJ97].

2.2.4 Cloud measurements

The Internet cloud measurements do typically measure packet delay and its variation as well as packet loss rates. Some examples of cloud measurement, i.e. measurements over long distances and between multiple locations are described.

2.2.4.1 Measurements for real time communications

An UDP echo tool was used in [Bol93] to measure delay characteristics between INRIA in France and University of Maryland in the USA. The transatlantic link bandwidth was 128 kbit/s at the time of the measurements (1992). The UDP echo packet includes three 6-byte timestamp fields: one for the sender, one for the receiver (echo) and one for the original sender as it receives reply. There is also a sequence number to detect losses. A phase plot was found convenient to express round trip time (RTT): for each packet round-trip time rtt_n, a marker is placed at position (x, y) where $x = rtt_n$ and $y = rtt_{n+1}$. If the delay is (relatively) constant, most of the points will reside at the diagonal. It was found that if the packet interval is short the delays are correlated. The conditional packet loss

is high which is quite natural. A queueing model describing this behaviour was introduced.

ATM wide area network (vBNS) was measured in [SSZ+96]. The delay and jitter for test traffic were analysed. It was found that the peak rate of a constant bit rate (CBR) connection rises as a function of the number of hops. The higher the original rate was, the higher was the relative rise in the rate.

The main reason for using the Internet to carry voice traffic is the price. However, there are also other reasons: silence detection saves much bandwidth and software based codecs may evolve more quickly and benefit more from the increase in processing power than ones built in the network. An experimental setup was described in [ML97]. There were three sending hosts (Rutgers University in New Jersey, University of California at Berkeley, and GMD in Germany) which all sent UDP packets (64 byte, 20 ms intervals equals rate 25.6 kbit/s) for 10 minutes each hour for a period of 14 days to the Bell labs in New Jersey. There were occasional problems in generating traffic so these incidents caused by end systems had to be filtered out.

The time was divided into distorted intervals and distortion free intervals. Distortion free interval was an at least 1.3 seconds period without errors according to studies of Brady in the 1960's [Bar68]. For some connections there were almost no distortion free intervals and these affected the average results much and made some figures hard to interpret. A moving average method over the preceding and the following hour was introduced to make the figures easier to interpret.

It was found that interstate connections provided good quality almost always but international connections provided bad quality when working days overlap. The authors suggest using the Internet to bypass local access network for interstate phone connections and bypassing the Internet using phone network for long distance.

2.2.4.2 Generic delay properties

As the TCP is a self-clocking transport protocol, it is sensitive to receiving acknowledgements properly [Jac88]. One of problems is the ACK-compression: a set of acknowledgement packets is delayed in a queue and their spacing is shorter when they arrive to the sender. In [Mog92] a possibility to identify ACK-compression is studied. It was found that it could be automatically identified. Furthermore, it was found that packet losses are correlated with ACK-compression.

One suggested topic for further studies was estimating the instantaneous RTT measuring time between sent data and the received acknowledgement.

The long-range dependence (LRD) is found to be a ubiquitous property of packet network traffic. Heavy tailed probability distributions are found from many network related statistics. However, the characteristics of the packet round-trip delay are not well studied even if they are very important for estimating a proper retransmission time-out (RTO) for TCP. Time stamp values for NTP packets were recorded between instrumented server and five other servers, two of them were in the USA, and in Australia, Sweden and Chile there was one server in each country [LM98].

Since restart could cause skew to results, a time period without server restarts was selected. The total number of packets exchanged between servers ranged from 11,000 to 53,000 packets, the average time between packets was 235 and 48 seconds, respectively. The Hurst parameter [LTWW93] was > 0.75 for all traces suggesting that packet delay is also a LRD process. Based on the traces it was found that the Kleinrock independence approximation might not be valid for the Internet. The authors think that *LRD in a packet round-trip delay process is caused by the LRD in the packet arrival traffic in the Internet.*

2.2.4.3 Large-scale probe measurements

V. Paxson carried out one of the largest Internet-wide measurements. The architecture was to install measurement daemons (network probe daemon, NPD) on different locations. A central management station controlled each probe. Based on instructions given by the management station a NPD contacted to another NPD and transferred a 100-kilobyte file. Each station captured packets on the network interface storing the header information and timestamp. The architecture made it possible to measure N^2 Internet paths with N probes. The number of probes varied, but was mostly around 35: resulting in more than 1,000 possible paths. For a detailed description of the system, see [Pax97c]. An improved follow-up project, NIMI, is described in [PMAM98].

In the measurements described in [Pax97b] Internet packet dynamics were studied: network pathologies such as out-of-order delivery, packet replication and packet corruption. Estimating bottleneck bandwidth was studied and "packet bunch modes" was found more robust than the packet pair. Packet loss ratio and characteristics of loss (burst and conditional loss) were studied. The system made it possible to measure and compare one-way packet delays with the round

trip times one would achieve with the simple "ping". The queueing delay and the available bandwidth were studied. Calibration of time values was discussed in [Pax98] and was done solely by analysing the results as NTP is not sufficient to keep millisecond resolution and GPS time is not always feasible.

Another result from the measurement traces was the development of an automated method to analyse TCP implementations and how they violate existing standards and good practices [Pax97a]. One of the most important parameters in TCP flow and congestion control is RTO; as its selection makes the difference between an unnecessary wait and an unnecessary retransmission. In [AP99] there is a comparison between different RTO estimators with actual traces. The sender based estimation algorithm is problematic as the ACK segments do not preserve the data segment spacing because of ACK-compression [Mog92]. The implementation of selective acknowledgement [MMFR96] and timestamps will help as they make it possible to use a more aggressive estimator for RTO. The data is based on the 1995 dataset and the authors propose a new analysis with newer data and live experiments.

Routing dynamics was studied in [Pax96] using traceroute, which is also a part of the NPD functionality. Routing pathologies were found: several routing loops of different duration (up to over 10 hours), one case of erroneous routing, rapidly oscillating routes, problems in underlying transport mechanism and temporary packet loss (for example because of congestion). If the most instable routes are not taken into account, one has over 90 % probability of finding a route that persists for a longer time than a week.

Routing tables were collected for ten months in [LAJ98] totalling over nine gigabytes of global (default-free) routing information. Local network provider provided their network status logs (obtained by pinging customers' network interfaces) and trouble tickets from network operations system. Also OSPF messages were collected. The paper focuses on two categories of failures: faults in the connections between service provider backbones and failures within provider backbone. It was found that availability was very low compared to PSTN (better than 99.999 %): only 30 % of routes had better than 99.99 % availability and 10 % of routes had worse than 95 %.

Frequency of routing updates was measured and peaks were found corresponding to intervals of 24 hours and 7 days in inter-operator routing. A similar pattern was not found in intra-provider routing. This suggests that much of the inter-operator routing updates are not caused by hardware or software faults but by failures in BGP [RL95] peering sessions at high loads.

2.2.5 Internet Protocol Performance Metrics (IPPM)

The IETF IPPM working group defines metrics for Internet performance. The framework document [PAMM98] defines criteria for those metrics, terminology, the metrics itself, the methodology, and the practical considerations including sources of uncertainty and errors.

According to the working group character:

> The IPPM WG will develop a set of standard metrics that can be applied to the quality, performance, and reliability of Internet data delivery services. These metrics will be designed such that they can be performed by network operators, end users, or independent testing groups. It is important that the metrics not represent a value judgement (i.e. define "good" and "bad"), but rather provide unbiased quantitative measures of performance.

As a full traffic analysis is not always feasible, the IPPM metrics are based on random sampling of the traffic. The framework document [PAMM98] includes a discussion recommending that the Internet properties should not be considered in probabilistic terms as there is *no static state* in the Internet.

So far the working group has produced the framework document [PAMM98] and metrics for measuring connectivity [MP99], one-way delay [AKZ99a], one-way packet loss [AKZ99b], round-trip delay [AKZ99c], and a framework for defining bulk transfer capacity metrics [MA01]. Several Internet Drafts are under process, such as packet delay variation metric, one-way loss pattern sample metrics, performance measurements for periodic streams and a protocol for one-way delay measurements.

2.3 Passive measurements

Passive measurements have the advantage that they do not interfere with normal operation of the network by increasing traffic into the network. The amount of data can be quite large: one 155 Mbit/s link can easily have an average utilisation of 80 Mbit/s: if one assumes that average packet size is 750 bytes, there will be about 13,000 packets per second.

The raw data rate is 10 mebibytes per second. High performance disks have a

capacity of 36 gibibytes: one disk can hold data worth of one hour. If only IP and transport layer headers (40 bytes per packet) are stored, the data flow is 0.5 megabytes per second: the same disk can hold over 18 hours of data.

There is quite a lot redundancy in the data that can be compressed away. All of the information is not always interesting, so when the measurements are planned, one should decide what information is unnecessary.

If one compares packet-based network measurements to those in a traditional circuit-switched network one sees a clear difference: to save essential information of a telephone call, 200 bytes are more than enough. This corresponds to 25 ms of speech at 64 kbit/s. If a typical telephone call is 3 minutes in duration, the record is only 0.01 % of data flow. For a packet switched network the header information is 5 % of data flow as the header is 40 bytes of an average of 750-byte packet.

2.3.1 Capturing data

The network technology used dictates how the data capture can be done. Measurements are simple when the network is a non-switched broadcast network such as IEEE 802.3 ("Ethernet") [ANS96]. Ethernet network adapters can be configured to a promiscuous mode to receive and forward to an operating system all frames seen in the network, not just the ones destined to its MAC address or to a broadcast address.

If the network technology utilises point-to-point links (like ATM, higher speed Ethernet networks, or serial lines), there are several ways to capture data.

Data copying in network node. Some networking equipment (like some OSI layer 2 (Ethernet, ATM) switches) can be configured to forward all packets seen in a port to another port, where those can be read. This may introduce some amount of jitter to packets.

Passive listening. Data on optical link can be copied using optical splitter that redirects some part of light signal to another fibre. The measurement equipment can be hooked at end of this fibre. As the optical splitter is a passive component, the measurement does not have any effect on normal network operation if the light budget has enough reserve capacity. A short break happens when the splitter is installed.

Similar methods can also be used on electrical links, but there may be some problems with high-speed links because the electrical characteristics of the

link may change.

Pass-through measurement device. The link is connected to the measurement device and the device copies incoming data to outgoing link verbatim. If the device is not working correctly, the network traffic will be disturbed.

The data collection can be either full monitoring (census) or sampling [AC89]. The latter is preferable over the first one because of economy, timeliness, large population size, inaccessibility of entire population, destructiveness of the observation, and accuracy. According to [AC89], an ideal sampling strategy has the following properties:

- Selects traffic frames at random with no preference for or avoidance of any particular class of traffic.
- Selects traffic frames as often as possible without interfering with other important tasks.
- Contributes a minimal overhead for monitoring.

2.3.2 Derived statistics based on captured data

While the shape of the distribution of a network parameter may be unknown, significant shifts can be identified by the Central Limit Theorem. Regardless of the shape of parent distribution, the mean of samples will approach the mean of the parent distribution and the standard deviation approaches the parent distribution standard deviation's divided by the square root of the sample size. Those statistics are easy to calculate as the mean and standard deviation can be calculated from three values: the count of samples, the sum of samples, and the sum of the squares of samples.

Some of the figures measured from the network are significant on their own while on other figures the change is significant. As the traffic will change over a period of time, the "normal" value will also change for many metrics. This can be automatically taken care by windowing observations. Authors of [AC89] suggest using *fixed-time based window with random sampling* as it does not require storing every sample from the window interval.

Traffic levels in (packet) networks are typically measured over time scales (15-30 minutes) which are quite long considering the burstiness of traffic. If the time

scale is shorter then there are problems with amount of data. In [EW94] some statistical descriptors are presented. These include peak-to-mean ratio (PMR), squared coefficient of variation (CSQ), correlation dimension (D_C), index of dispersion for counts (IDC), peakedness and the Hurst parameter. The first three (PMR, CSQ, D_C) were considered as practical measures and studied further with traces from an Ethernet network (6 traces, 100,000 packets each) and ISDN signalling network (5 streams). It was found that PMR could provide information on capacity exhaust, if the time interval is carefully selected.

Analysis of self-similar queueing performance was presented in [ENW96]. Both Ethernet and ISDN traces were used. A Queueing Network Analyser (QNA, GI/G/1 approximation with two moment characterisation) was fitted on trace and average delay with different utilisation delays was compared. The curves differ at high loads: the trace jumps utilisation exceeding 0.5 while QNA rises slower at 0.8. The trace was shuffled randomly thus destroying all correlations but still maintaining the marginal distribution. This curve agrees with the QNA curve.

At the second step the trace was divided into fixed size blocks (10-100 packets), thus preserving local bursts. Using block size of 25 packets the blocks were shuffled externally (order of blocks is changed but the order of packets within a block was maintained). The queueing performance differed from the original trace. When the trace was internally shuffled (order of blocks is maintained but the order of packets within a block is changed), the trace preserved long-range correlation and agreed with the original trace in queueing performance. The FBM was fitted to the trace with the exception that under time scales of 10 milliseconds the short-range correlations dominate. Also it was seen that due to the finite length of the trace, the queue length distribution from the trace for large values would fall off faster than predicted by the model.

It was concluded that high time-resolution traces are not practical in operational engineering. As some systems can report traffic counts over intervals longer than one second, those figures may be useful for estimating the Hurst parameter.

In [KqL96] it is proposed dividing traffic in frequency domain into three categories: low-frequency ($|\omega| \leq \omega_L$), high-frequency ($|\omega| \geq \omega_H$) and mid-frequency ($\omega_L \leq |\omega| \leq \omega_H$). The peak rate of the low-frequency traffic defined link bandwidth while buffering has most effect on high-frequency traffic. Proper selection of ω_L and ω_H will help practical measurements and analysis.

2.3.2.1 Backbone measurements

A LAN-to-LAN interconnection service using DQDB MAN was measured in [CMGR94] for four hours (11:00-15:00) on a workday. The measurement confirmed findings from earlier [LTWW94] studies. The IDC for traces (mostly TCP on top of IP) was calculated. It was found that a two-stage Markov modulated Poisson process (2s-MMPP) is suitable for short time analysis and the fractional Gaussian noise (FGN) for longer time scales.

The NSFNET backbone was measured in [CPB93] for characteristics described in Section 2.1. The total traffic volume and the trend as well as the protocol distribution were measured. Daily variations of packet sizes were found corresponding to utilisation rates as bulk transfer applications are run off-peak hours. However, no long-term trend for packet sizes was found. Few sites (31 of 4254 networks, 0.7 %) generated half of the total traffic and 118 (2.8 %) sites received 50 % of traffic. Furthermore, 46.9 % of traffic was between 1,500 site pairs (0.28 % of 560,049 possible).

In [CBP95] a time-out mechanism was introduced to define a flow. As the natural definition of a TCP flow would be one delimited with SYN and FIN segments, this is not feasible for core network measurements for following reasons:

- The measurement equipment may be drop some packets.
- One of the end stations may become unavailable (either end system or its network connectivity) and never sends a FIN segment.
- Route may change during a flow and the rest of flow is never seen at the measurement point.
- SYN/FIN applies only to TCP flows, for UDP it is not possible to know in the network where a flow starts and where it ends.

A flow was defined as an unidirectional packet train [JR96] and a flow can be defined with desired granularity starting from the address-port tuple and ending to one including traffic between a large number of networks. Analysis was based on one-hour measurements both in the NSFNET backbone and campus networks. Different time-out values ranging from 2 to 2048 seconds were studied. Time-out defines the maximum time between two successive packets to be considered belonging into the same flow. Many protocols such as DNS [Moc87] and data transfer of FTP [PR85] were unaffected by the selection of the timeout while

others, such as telnet [PR83], were greatly affected by it. See Section 5.1 for further discussion about flows.

The network protocol and the packet size distribution and flow characteristics measured from the vBNS network with OC3MON can be found from [TMW97]. This measurement suffers from the fact that flows are artificially expired once an hour because of a limitation of OC3MON, which will affect statistics for long-lived flows such as multicast traffic.

2.3.2.2 Local area network measurements

A local computer network was studied in [Dra92]. Workloads for different login (local-remote) and disk (local-remote) combinations were measured using tcpdump. Application benchmark response time, throughput and utilisation of shared resources were measured. The system bottlenecks were found and some ways to improve performance were studied.

Traffic from local area network was collected for 5 hours and the trace collected with tcpdump was analysed to characterise different applications [BS92]. The results include average rate for each minute and the percentage of different protocols over the measurement period, as well as the packet size distribution.

In the work described in [ADPCH+92] University of Florida Ethernet backbone was measured. The results include the observation that the network load was about 3.7 %, 4.3 % and 6.9 % over the busiest hour, 30 and 10 minutes, respectively. Furthermore the hourly traffic was divided into five categories depending on the volume of the traffic. Packet size distribution was found to be constant over a 24-hour period with the average size of 138.6 bytes.

2.3.2.3 Dial-up measurements

Paper [CE97] discusses dial-up session call measurements carried out for 30 days totalling about 500,000 calls. Following parameters were recorded from each call:

1. call starting time,
2. call duration,
3. total number of information bytes and packets transferred from a user to the network,

4. total number of information bytes and packets transferred from the network to the user.

Based on the average packet size from the network to the user the calls were divided into two classes: A and B. The average call holding time in class A was around 300 seconds while in class B it was around 1,700 seconds. Also the direction of the flow was different: in class A the bit rate from the user to the network was higher. The findings suggest that class A calls are for sending and checking email while class B calls are for web surfing and similar tasks.

The call arrival process was also characterised and it was found that the call interarrival time could be modelled by a hyperexponential distribution as it is the sum of a range of exponential distributions.

2.3.3 Simple statistics

Network devices do have various counters for monitoring and statistics collection purposes. These counters include number of packets and bytes transmitted and received, number of errors and so on. These data can be queried using a network management protocol such as SNMP [WHP99].

2.3.3.1 Application traffic

HTTP traffic was measured from an Ethernet network in [Mah97]. The total duration of traces (4 different) was about 10 days and total 1.5 million packets were captured. Based on measured data the following quantities were modelled: request and reply lengths, document size, think time (user time between pages), consecutive document retrievals (number of documents from a single server) and server selection. Periodic retrievals by a webcam every 5 minutes were removed from traces as they skewed results. A simple time-out heuristic was used to decide whether a connection was part of the current page or if it was a new page.

Modem user traffic at Internet Service Provider was measured twice for an hour (in 1997 and 1998) in [FGHW99]. The traces are analysed for scaling behaviour using wavelet-based analysis. The second trace was further divided into three subsets: first consisting of local traffic (modem user – ISP web server), second remote (modem user – major news servers), and third realaudio (UDP) traffic. The ns-2 simulator was used for replicating realistic HTTP traffic to study TCP dynamics. The authors propose that analysis techniques can be used to find potential

performance problems (bottleneck links, misbehaving connections) in real-time. However, the feasibility of practical implementation is unknown. Another conclusion was that *by relying almost exclusively on the physical or networking-related understanding of the impacts of the various user- and network-related aspects of variability and of such basic concepts as closed-loop flow control, it appears to be possible to end up with a full-blown networking environment that is in the right "ball park" when compared to real networks.*

The log files for electronic mail system were studied in [PS89]. No distribution of the standard families was found to be a good match for the message size distribution because there was a high peak around 300 bytes. The authors propose that the distribution may be a sum of several distributions for different types of users, but this was not verified. It was found that exponential distribution did not fit for the interarrival times of messages (using χ^2 test). The Weibull distribution was a better fit so the message arrival process was not Poisson.

For many interactive applications such as graphical design tools, it is difficult to obtain traffic parameters *a priori*. SNMP measurements were used to estimate *equivalent bandwidth* for existing connections in [CCH95]. This measured equivalent bandwidth was then used to estimate if there is enough capacity to support the application in question.

Using on-line measurements to support call admission control (CAC) was studied in [SW98]. A network monitor measures traffic and maintains a histogram over a few last minutes. Based on that the system predicts available resources and estimates whether the new connection can be accepted or not.

2.3.4 Measurement organisation

The measurement can take place in a single location or it may take place in several locations. An example of a measurement setup is presented in Figure 2.4. There are two passive measurement devices attached to two locations in the network. The equipment monitors network data and records packets destined to another location in compressed form (see Section 2.3.5) with a timestamp. For discussion about timing accuracy, see Section 2.5.1. The traffic to be measured may be a part of normal data or data sent only for measurement purposes.

After measurement has taken place, the measurement data (fingerprints and timestamps) are exchanged either on-line or off-line (tapes, CD-ROM) and post-processing of data is done [GD98].

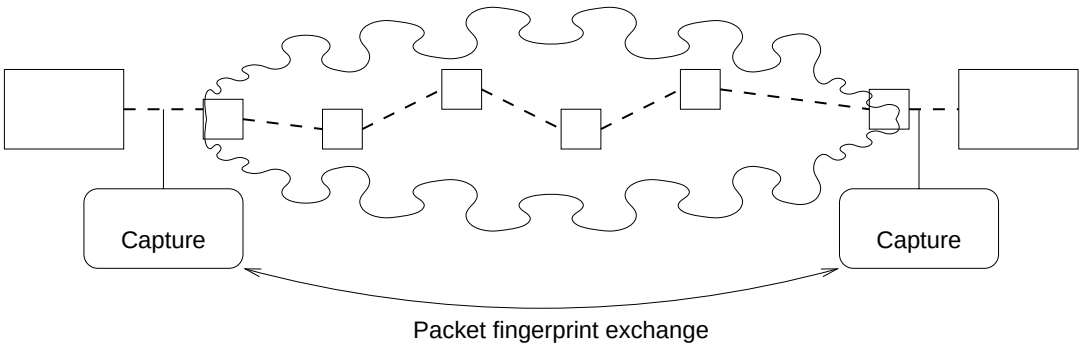


Figure 2.4: Traffic capture at end points.

It is also possible to locate measurement software on end systems. There are, however, some factors that may cause problems in this case. It also has some benefits; some discussion can be found in [Pax97b].

2.3.5 Trace compression and fingerprints

As noted above (Section 2.3), the amount of data can be much too large to be handled as-is. Some compression is needed to keep the amount of data manageable.

2.3.5.1 Packet and flow fingerprinting

One possibility to reduce the amount of data to be stored is to utilise packet or flow fingerprints. This can be done if one is not interested in the actual content of a packet but wants to locate the same packet in different parts of a network. These studies include delay and loss studies, routing studies and tracing of DDoS attacks.

One calculates a digest from a packet using some algorithm. While simple summation of all packet octets may be sufficient in some cases, a high collision ratio may result. The use of cryptographic strong message-digest algorithms such as MD5 [Riv92] and SHA [Ano95] should produce a low collision ratio even if only a part of the hash value is used. An intermediate solution between processing costs and low probability of collisions would be using a suitable CRC polynomial as all properties of cryptographic message-digest algorithms are not needed.

One must take into account that some fields of the IP header (Figure 2.5, p. 39), for example, are volatile. These include DS byte, TTL and check sum fields. These parts should be ignored from digest computation.

Similar approach can be applied to documents: one may be interested if the same document is transferred multiple times over the same link. If this is the case, then caching may benefit users. One has to reconstruct application level data from distinct packets, which is not always a trivial task [Fel98]. Additionally, the connection may include some volatile data such as the current time that should be masked out when calculating the message digest.

2.3.5.2 Flow-based compression

As each connection typically has several IP packets, the subsequent packets have very similar headers. For example, in a normal UDP or TCP transfer there are typically only few fields that change:

1. IP datagram identifier, needed in the case of fragmentation. Modern implementations try to avoid TCP segment fragmentation by PMTU discovery [MD90].
2. IP header checksum.
3. TCP sequence number.
4. TCP acknowledgement number.
5. UDP or TCP checksum.

The datagram identifier is useful in some cases (like locating duplicated datagrams), but in source or flow modelling it is of no use. There was much discussion about the role of the IP identifier on end-to-end interest mailing list in May 2001 [End]. The datagram identifier is obsolete if datagram fragmentation in the network is disallowed as is case with new TCP implementations and IPv6. The checksum fields are computed from the data; the IP header checksum can be verified in any case and the TCP or UDP checksum if the whole datagram is captured.

If there is a route change in the network along the path datagram travels and the number of hops changes, the TTL field will change also. If one wants to monitor possible route changes, the TTL value must be recorded.[1]

[1]One must note that the route may have been changed even if the TTL value does not change if both routes have same number of hops.

The sequence and acknowledgement numbers of subsequent packets belonging to the same flow are typically very close to each other: if there is no packet reorder or loss both are increased by the payload size in each packet. Typically, one can compress headers to the ratio of 10:3 or better [Jac90, DNP99]. The compression ratio can of course be improved with normal data compression. Flow compression is studied in Section 3.1.1.1.

2.4 Measurement tools

Over time, multiple tools have been developed for network measurements, often as an ad-hoc solution to resolve a current problem in a network. Tools are later developed further to be more generic or easier to use. An advanced tool may utilise information from multiple sources such as IP address allocation databases in addition to direct measurements.

The following survey has used The CAIDA Internet Tools Taxonomy as the primary list of tools [Cai]. Only a small set of tools is discussed here, based on the author's selection which ones are important considering this work. For the full list, see [Cai].

2.4.1 Tools for availability and delay

Probably the most widely used network measurement tool is ping. It is readily available on most systems. The basic version[2] sends ICMP echo request packets and for each received echo reply it prints a line indicating packet sequence number and elapsed time for each received packet. At the end it prints a summary line indicating minimum, mean and maximum delay and packet loss rate.

Timing accuracy is dependent on the host operating system and if kernel-level timestamps are supported. A typical resolution is around one millisecond on modern systems. Variants that support parallel probes, stress testing or graphically visualised output have been developed

[2]Available from ftp://ftp.arl.mil/pub/ping.shar

2.4.2 Hop by hop characterisation tools

These tools are based on sending an UDP datagram to (hopefully) an unused port[3] with varying TTL values in IP datagram. When a router decrements the TTL field and finds it to be zero, it sends back an ICMP time exceeded message. For this message the program will learn the next hop on the route and sends another packet with the TTL field incremented by one. When the message reaches the destination and there is no listening process in that UDP port, the host will reply with the ICMP port unreachable message.

The original traceroute[4] has several variants that have some performance improvements or do display some more information such as Autonomous System [RL95] numbers. Some variants use ICMP echo request messages instead of UDP packets.

2.4.3 Throughput measurement tools

There may be two different objectives for throughput measurements. The first one is to measure instantaneous available throughput under the present network conditions and the other one is to measure maximum achievable throughput in absence of competing traffic. The first one is used to estimate the application throughput and the latter one is used to characterise network and network equipment.

Throughput measurements can be obtrusive if the measurement tool does not perform similar rate adaption as application protocols. If a measurement uses a real application protocol, only a relatively large number of simultaneous measurements distract other network users.

2.4.3.1 Synthetic throughput measurements

A modification of traceroute, pathchar[5] [Jac97] uses several (default 32) probe packets with different sizes to measure the link line rate hop-by-hop. The program (no source code published) gives fairly good estimates (correct within an order of magnitude) in spite of that the estimate is based on small time differ-

[3]The default port is $33434 + n_{hops}$.

[4]Available from ftp://ftp.ee.lbl.gov/traceroute.tar.Z, variants from ftp://ftp.nikhef.nl/pub/network/.

[5]ftp://ftp.ee.lbl.gov/pathchar/

ences (around 80 µs for typical links) in data where differences between samples are much larger (10–500 ms). Pathchar and its results were analysed in [Dow99]. Some improvements are suggested for calculating the estimated link bandwidth improving both the accuracy of the estimate and reducing the number of probes needed.

The cprobe [CC96b] can be used to measure the available bandwidth for a greedy application (with no flow control). The TReno is a similar tool but implements flow control: the results should be the same as the ones obtained with TCP transfer with up-to-date implementation of TCP [Tre00] with normal limitations of non-application protocol tests (see Section 2.2.2).

The NetSpec is a scripting language for the design of communication and workload pattern throughput performance measurements. It includes models for CBR, telnet, FTP and WWW traffic [LFJ97].

2.4.3.2 Transport protocol throughput

Bulk transfer tests using a benchmark application are very useful for comparing different hardware and operating system performance. These tools may provide, in addition to throughput figures, extra information such as CPU utilisation metrics. There are also a large number of settings to control buffer sizes and other parameters, which do have an effect on the end system throughput. These tools include ttcp, nttcp, netperf and DBS [Muu, Bar94, Net95, SPI97].

2.4.4 Packet trace collection

The most popular packet collector is tcpdump, which uses a highly portable library libpcap to capture packets, which hides operating system differences from the capture software. Tcpdump is available[6] for multiple operating systems, typically for different UNIX variants. There are other tools for different operating systems, but tcpdump is the de-facto standard in research.

Trace files written by tcpdump, "pcap", are a common format to exchange packet traces. It has, however, some limitations in its format. The files are not always self-contained but some auxiliary data must be provided to properly analyse data and timestamps have a limited accuracy, for example. To solve these problems a successor of OC3MON [ACTW96], CoralReef project, has introduced a new file

[6]Available from http://www.tcpdump.org.

format, which should allow accurate recording of packet traces [Cor]. CoralReef also supports a variety of special hardware cards and has a common API to hide implementation details from analysis programs.

There are also a large number of hardware-based measurement devices. These equipments have their focus either on benchmarking or network operations problem solving and they are not capable of sustained large volume captures from networks.

A flow-level data is generally sufficient for accounting purposes. The NetFlow Interface, supported by Cisco routers [Neta], exports flow records from routers. There is on-going work on IETF to standardise IP flow information export [IPF02].

2.5 Issues to be considered in measurements

There are some problems in measurements in addition to the ones described above. These problems are caused by deficiencies of equipment, policies of organisations and legislation.

2.5.1 Timing accuracy

If there are multiple measurement points and measurement results include wall clock time, a problem for clock synchronisation arises. Network time protocol (NTP) can be used to synchronise node clocks from a reference clock but there are reasons why it is not suitable for network measurement clock synchronisation [Pax97c, p. 185]:

1. NTP focuses more on long-term accuracy at the expense of short-term skew and drift.
2. As the time information is transported in IP network, it is also subject to delay variations and thus can cause hard-to-solve problems in results.
3. Two computers in NTP peering can synchronise their clocks to approximately 10 ms that is not sufficient for high-resolution delay measurements.

The accuracy of a real-time clock in computers is not very good: typically error can be several seconds in a day. For example, if the error is 5 seconds a day, in

a 10-minute period the error is 35 milliseconds, which is in the same range as network delays. If one is measuring one-way delay, the results may be severely errored even after a short measurement period, and totally useless for persistent measurements.

If definitive accuracy is needed, the timing information must be provided out-of-band. There are several timing sources that transmit radio signal, but the most useful source for timing information currently is GPS. The SPS available for civil users provides timing information with 350-nanosecond accuracy for 95 % of time [Dan99]. Several commercial modules are available; the low-cost ones provide $\pm 1\,\mu s$ accuracy for 1 PPS signal.

2.5.2 Privacy issues in measurements

When measurements are taken from an operational network, privacy issues arise. The network traffic carries potentially sensitive information such as passwords or other identification information. While common use of encrypted connections (IPSec, TLS, and SSH) protects the payload, even the knowledge that there exists a communication between two parties can be sensitive.

The legislation varies from one country to another and it is not always clear if something is permitted or not as legislation does not usually refer to a particular technology but speaks in generic terms. The Finnish legislation, for example, is quite strict against even inadvertent *unauthorised* wiretapping. Revealing that there has been communication between two parties can result in a fine or even imprisonment.

US federal code 18 U.S.C. § 2511 "Interception and disclosure of wire, oral, or electronic communications prohibited" prohibits wiretapping in general but allows it for an operator, if needed by network operations, or by a court order.

2.5.2.1 Finnish legislation

In the Finnish legislation, two laws and one statue protect privacy of communication, all updated in 1999 or 2000:

- Penal law, Chapter 38 about data and communications crimes[7]

[7] Rikoslain 38 luku, tieto- ja viestintärikoksista.

- Communication law 121 about privacy protection in telecommunications and data security in telecommunications[8]
- Communication statue 121a about privacy protection in telecommunications and data security in telecommunications[9]

The Communication law 121 defines among others the following terms:

Person identification. All information describing a person or one's properties or living circumstances. Based on this information a single person, one's family, or people living in a same household can be identified.

Teleprovider. A person or a legal entity that provides teleservices for other public.

Subscriber. A person or a legal entity that has made agreement with a teleprovider to be able to use services provided by the teleprovider.

User. A person who is using teleservices.

Identification information. Subscriber or user's number or other identification generated or stored in establishing teleconnection.

The basic principle in the Finnish legislation is that if a message is not intended for public, it is confidential. If a non-intended recipient receives a confidential message, one may not indicate the content or existence of the message or utilise the information in the message without permission (Cl 121 4 §).

The reasoning for not making the reception of a non-intended message criminal is that one may unintentionally receive a private message. A marine VHF system is an example of such system where one may hear a private communication by change. If a message is protected somehow, for example with encryption or with a physical envelope, it is prohibited to remove the protection (Pl 38 3 §).

2.5.2.2 What is sensitive in Internet protocols

If one looks at the IP header (Figure 2.5), most of the fields are non-sensitive. There are only two fields that carry sensitive data: the sender's address and the

[8]Vi 121, laki yksityisyyden suojasta televiestinnässä ja teletoiminnan tietoturvasta.
[9]Vi 121a, asetus yksityisyyden suojasta televiestinnässä ja teletoiminnan tietoturvasta.

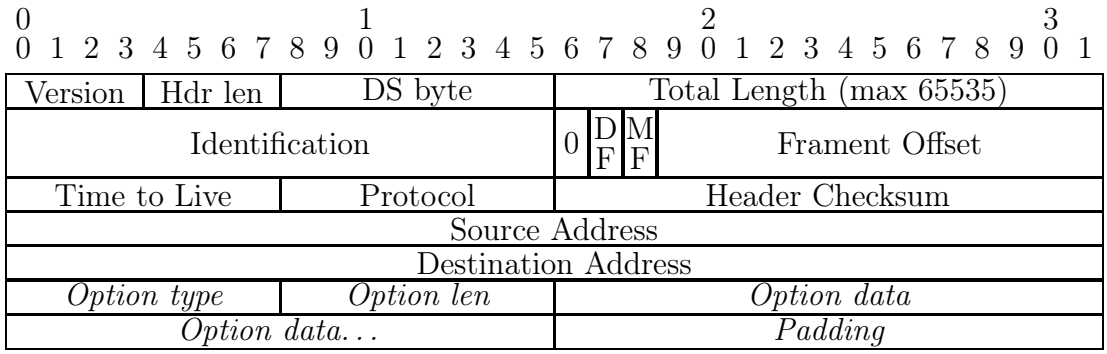


Figure 2.5: IP datagram header structure [Pos81b].

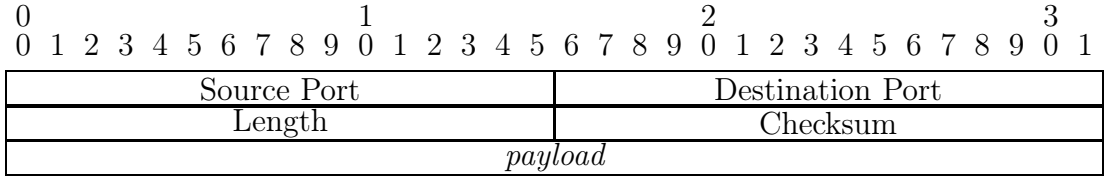


Figure 2.6: UDP header format [Pos80].

recipient's address. They identify the communicating parties to the host granularity. In many cases, this is the same as to a single person and thus it is a *person identification*.

The check sum field can be sensitive, since if all other fields are known except the address fields, it is possible to rule out some set of possible IP addresses. Based on the TTL value, it is possible to guess how many hops the IP packet has travelled as implementations use values of 32, 64, 128 and 255 for TTL. The total length of an IP datagram can give some information about upper protocols. However, the information in these fields cannot be considered sensitive, as there are many possible matches.

The UDP (Figure 2.6) port numbers are used to identify the application. Based on the information one can answer the question: "is somebody using certain application in this network". Again, it depends on the number of users in the network whether this information is sensitive. The UDP checksum may reveal something about payload data if short packets are used and all of the source and destination IP addresses and UDP port numbers are known. Those are used in the pseudo header to calculate the check sum.

The same discussion about port numbers and check sum applies also to TCP (Figure 2.7) as they have the same functionality. Other fields in the TCP header are related to connection setup or flow control and do not reveal any other in-

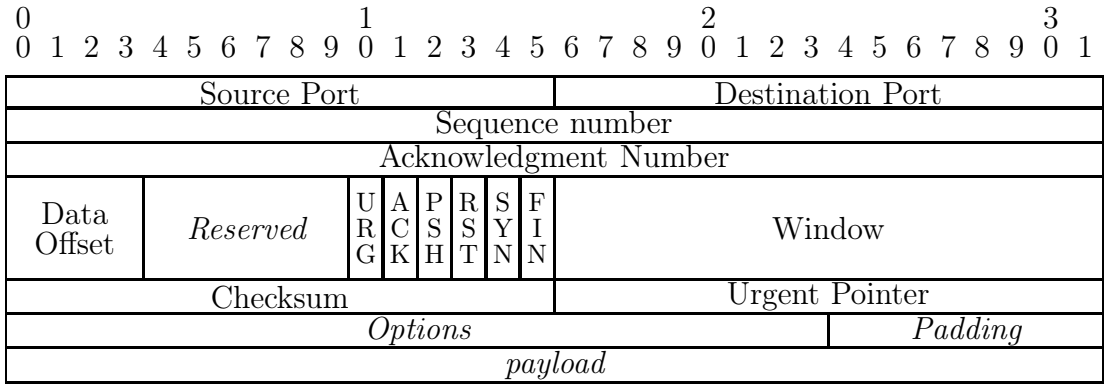


Figure 2.7: TCP header format [Pos81c].

formation than bytes transferred over connection. A pointer for urgent data may carry some application semantics.

The payload carried in UDP or TCP packets can be considered sensitive as the application data may contain for example private email. This information is clearly protected by legislation.

To conclude, both the IP addresses and the payload are sensitive information. The check sum field in short packets may also contain some sensitive information.

2.5.2.3 How to protect privacy

The destination and source IP addresses potentially identify (with the accuracy of a single person) the communicating parties. To work safely, the IP addresses should be sanitised. There are several possibilities to remove sensitive information:

- Each network address is replaced with a random number. This makes it practically impossible to trace a single user in a large network. The problem is that topology information of the network is lost.
- Lowest-order byte of the network address is replaced with a random number. This will protect a single user in most cases and still preserves routing information.
- Routing prefix is maintained intact, subnetwork and host parts are replaced with a random number. This also maintains the network topology but allows a greater level of privacy as the route prefix can be quite short.

  This, of course, requires that routing information (BGP or similar) is available for the measurement device. One can also use a list of autonomous

systems.

2.6 Conclusions

Traffic measurements can be either passive or active. Active measurements measure the state of network at that moment while passive measurements measure either end system characteristics or a combination of network dynamics and end system functions.

There are several measurement efforts underway: some co-operation is needed as the traffic mix and network characteristics can be very different in different parts of the Internet. Also specialised measurements need system setup at both ends of a connection to measure both directions of data flows. Time synchronisation is also a demanding task. The GPS system provides a good timing reference but it needs specialised hardware.

The IETF IPPM working group is doing work to establish common metrics and methods to measure network performance. The ITU-T is also working for similar metrics [ITU99].

Efficient handling of measurement data is one of the challenges for Internet measurements; especially for traffic capture.

In many papers the measurement period has been quite short, especially in papers where some stochastic model fitting had been done. Following metrics have been proposed to be calculated from long-run measurements:

- Protocol and application distribution, traffic volume by day, date and direction like in [TMW97].
- Hurst parameter estimation using different statistical methods [JW97].
- Traffic measures to identify the peak rate [EW94].
- Estimating instantaneous RTT measuring time between sent data and received acknowledgements [Mog92].

Chapter 3

Network measurement setup

The practical part of this work consisted of two tasks: to implement a measurement system and to capture protocol header data for analysis. The choice of an equipment is dependent on technology used on the link to be monitored. The link between the Networking Laboratory and the rest of the campus of Helsinki University of Technology was chosen as it was easily accessible and measurements on it did not need any special agreements or arrangements. The technology used on the link was ATM on optical multimode fibre at the speed of 155 Mbit/s. The IP protocol was carried over LAN Emulation [Com97] emulating Ethernet.

3.1 Measurement organisation

There were approximately 30 to 50 users – mainly researchers – in the laboratory using the link as the access to the Internet and campus services from their LAN-connected PC's and workstations. There are also two public web servers in the laboratory, which are mostly accessed by students as they provide course information and material. The web servers have also off-campus users as there are hundreds of student reports, which have topics of public interests. In the network there are also email servers for laboratory personnel and some smaller services. It is thus possible to thus monitor both the access and the server ends at the same location.

3.1.1 Measurement equipment and software

The measurement system is Intel-based PC (IA32) with 256 MiB of memory and 4 SCSI3 disks. Measurement cards are DAG3.2 cards by Waikato University (NZ) [DAG]. The cards are set up to capture two first cells of every protocol data unit (PDU). Two-cell capture is needed to record TCP flags in the LAN Emulation network [Com97, p. 51] [Pos81b, Pos81c].

The card transfers cells with timestamps to a central memory where they are available for the software that reassembles the PDUs. As the PDU boundaries are not preserved if there are more than two cells in a PDU, a simple logic is used to keep conversion synchronised in the PDU stream. Each cell expected to be the first cell of a PDU is checked if it contains a valid IP header. An IP header is considered to be valid if and only if:

1. IP version equals 4.
2. Header length is equal to or greater than five 32-bit words.
3. Header length is less than nine 32-bit words.
4. Check sum for a header is valid.

It is also checked that the cell is not the last cell of the PDU as it might be in the case of a short UDP packet. If not, then another cell is waited for before the reassembled packet is passed to compression routines.

3.1.1.1 Flow based compression

Network traffic capture with current link speeds needs much storage. For network monitoring aggregate metrics can be calculated and data reduced even further as needed information is known in advance. For research purposes, it is not always known what information is important. If possible, all of the header data is saved. However, there are many fields in the IP, UDP, and TCP headers that do not change over the lifetime of a connection when observed at a single location, as described in Section 2.3.

The method described in [Jac90, DNP99] is to code only the difference between consecutive packets in same flow and use a short code for default (in-order delivery) case. The method is intended to be used when the number of connections

is small, e.g. on first-hop links. However, a larger identifier space can be used to identify more simultaneous connections. The source and destination IP address, the protocol number and, in case of UDP and TCP, also the port number, define a flow. These values are used as input to a pseudo random number generator (PRNG) whose result modulo a prime number P is an index to a table of size P, effectively a hash table.

The packet is compared with the one in that table position. If the packet belongs to a different flow than the one in the table entry (hash collision, or previously vacant position), the packet is inserted into the table and a new flow record (packet headers as such) is written to the output stream and the packet is saved in the table. If the packet belongs to the same flow, the difference between these two packets is identified. If the packet is:

1. TCP packet, then check if other TCP fields (except the check sum) are the same and if it is the one of the following cases:

   (a) $\Delta\text{seq} = \text{databytes}_{n-1}$ (in-sequence), $\Delta\text{ack} = 0$, and data length equals $\Rightarrow$32 bits

   (b) $\Delta\text{seq} = \text{databytes}_{n-1}$, $|\Delta\text{ack}| < 2^{15}$, and data length equals $\Rightarrow$48 bits

   (c) $\Delta\text{seq} = \text{databytes}_{n-1}$, $|\Delta\text{ack}| < 2^{15}$, and data length differs $\Rightarrow$64 bits

   (d) $\Delta\text{seq} = 0$, $|\Delta\text{ack}| < 2^{15}$ $\Rightarrow$48 bits

   (e) $|\Delta\text{ack}| < 2^{16}$, $|\Delta\text{ack}| < 2^{15}$ $\Rightarrow$80 bits

   If the flags were different, or sequence or acknowledgement numbers were out of 32 KiB range, then all of the datagram is stored. TCP options are saved as such as there is no efficient way to compress SACK [MMFR96] or timestamp [JBB92] options. The EOP and NOP options [Pos81c] are removed.

2. UDP packet, check if the size is the same in both

3. ICMP packet, check if the type, code and possible extra fields are the same

4. IP-in-IP: save flow info and find a flow for the encapsulated IP packet

5. other packet: save the IP header

Time information with a microsecond resolution is stored with variable-length coding: if the time difference with the previous packet (not necessary in the same flow) is:

- $< 2^{15}$ µs (32 ms) ⇒coded with 16 bits,
- $< 2^{31}$ µs (2 s) ⇒coded with 32 bits,
- else ⇒coded with 96 bits.

If a finer granularity is needed, it is trivial to change the base unit to nanoseconds, in which case the limits change proportionally.

In an optimal case, where an in-sequence TCP segment arrives within 32 ms of the preceding packet and no hash collision happens, the packet is stored with 48 bits, i.e. 6 bytes in contrast to 48 bytes without compression. Of course one can also use general-purpose data compression. The non-flow compressed file yields a better compression ratio because of greater redundancy, but still the advantage is on the order of 1:2.

3.2 How to sanitise network addresses

There are several approaches to solve the privacy problem described in Section 2.5.2. One commonly used method is to replace the IP addresses in network traces with sequentially allocated numbers. A table is maintained for mappings between real and fake addresses. Once the remapping is done, one cannot know which fake address corresponds to which real address. This approach has, however, several drawbacks:

- Topology information is lost.
- There is no mapping between subsequent traces unless the table of mappings is stored *securely*.
- It is not possible to correlate traces from different points of network.
- The table may grow large and thus become difficult to store, especially in a measurement device.

To overcome these problems, a new cryptography based solution was designed. A brute force search over the whole data set is feasible because there are only a limited number of IP addresses. The algorithm must be selected carefully for this reason.

There are several ways to generate the encrypted value. One possibility is to use keyed-hashing [KBC97]: concatenating a secret value with the IP address and calculating hash value over this data. If the hash function is cryptographically secure it is not possible to find out the IP address provided that the secret is long enough to prevent a brute-force search. Another possibility is to use data encryption. A symmetric encryption was selected because of faster operation in comparison with keyed-hashing or asymmetric encryption.

3.2.1 A solution to sanitise network addresses

The system takes at maximum a 1024 byte secure key, which can be read from a file or from a stream. Reading from a stream is recommended in order to avoid the secure key being written to a disk if the secure key is protected with encryption such as OpenPGP [CDFT98]. If a single trace without mapping between different traces is wanted, `/dev/random` or a similar source of random bits can be used as a key. A 128-bit key for Blowfish [Sch96, p. 336] is generated using MD5 over the supplied key.

Each time an IP address is seen in a packet, either in the IP header or in some other location, such as in data part of an ICMP message, it is scrambled. A table similar to a routing table is then consulted to find "hostpart", i.e. how many bits from the address should be scrambled; the "network part" is left unencrypted. The original IP address is concatenated with a 32-bit block of the key and encrypted into a 64-bit value using Blowfish in ECB mode. As the encrypted addresses should be evenly distributed, the low part of the encrypted value is used as an index to the hash table. If there is no entry at that location or the entry is different, a special token is written into the packet output stream with the top 24 bits of real address[1] and encrypted value.

The original IP address in a packet is replaced with a value that has its topmost 8 bits from the original address and lowest $\lceil \log_2 \mathit{tablesize} \rceil$ bits from the encrypted value. A new record is written onto the stream in case of hash collision. If the table is large enough, the collisions are expected to be rare. There is a possibility that two addresses in the same IP packet produce the same hash value but this can be resolved in decoding.

When the compressed and scrambled stream is expanded, the encrypted values are used as a key to the database where the random IP addresses within each

[1] The network part can be shorter than 24 bits, in that case lower bits are zero.

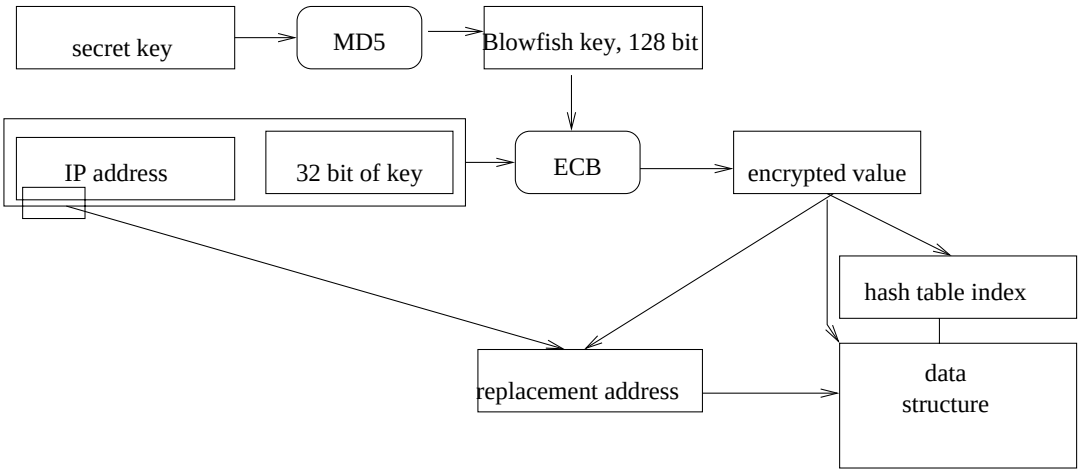


Figure 3.1: Scrambling of IP addresses.

byte	contents	description
0	Enc|10.2.3.0|71ee1331 ...	IP address scramble info
12	Enc|192.168.0.0|b64af064...	IP address scramble info
24	t=971702544.237632|(ip+tcp)	IP packet
86	Enc|192.168.0.0|4af0b664...	IP address scramble info
98	t=971702544.246370|(ip+tcp)	IP packet

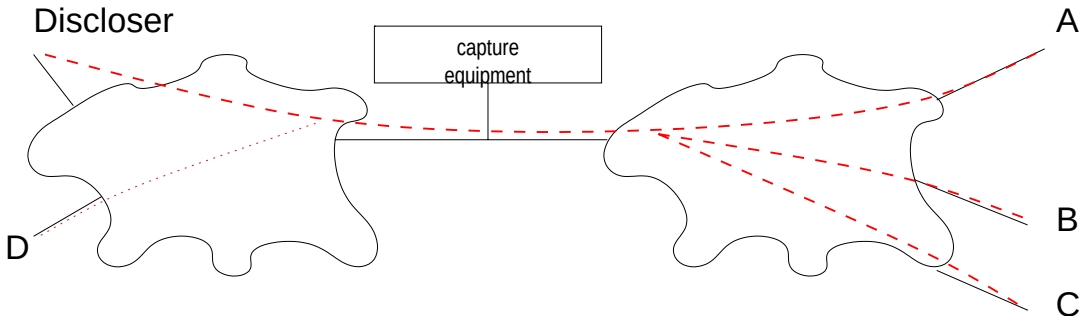
Figure 3.2: Example of trace file contents with encrypted IP address.

network are generated. If there is no database entry for the encrypted value, a new free random IP address within the network is selected and used as replacement for the encrypted value. A single encrypted IP address maps always onto the same replacement value as the database is disk based and persistent. This makes it possible to keep one-to-one mapping between different traces and there is no need for any database functionality in the measurement device. This eases performance requirements for the measurement device, which can be seen as a filter that takes IP traffic and produces a privacy-protected stream of packet headers. An example of the file contents is shown in Figure 3.2.

The database used for one-to-one re-mapping of IP addresses contains no sensitive information and thus does not need any special handling. The only data that must be protected is the secret key used in the measurement device.

3.2.1.1 Possibility to address disclosure

The address scrambling has one weak point, namely it is vulnerable to chosen plain text (in this case chosen IP address) attacks. If the Evil knows that there is an on-going measurement and that the results will be published, one can find out mapping between selected IP addresses.



1. Send packets from Evil to A, B and C with some identifiable pattern.
2. Send packets from Evil forging D as source address to A, B and C with some identifiable pattern.
3. Locate sent packets from trace by taking into account that some packets may be lost.

Figure 3.3: Chosen IP address attack.

The Evil sends some, possibly handcrafted, IP packets in some defined pattern; as the port numbers are left intact, one may utilise them to carry extra information. The aim is to be able to locate packets from scrambled and published data. As one locates the packets (maybe using timestamps as a helper) the value corresponding one's IP address is found.

When the Evil has supplied enough packets to learn its own IP address, one can send packets to those hosts one wants to learn about. If this is done in some redundant pattern, the sender can learn any IP address, if the packets to the wanted destination pass the measurement point. It may also be possible to use forged sender address to learn IP addresses on the same side of the measurement point unless strict network ingress filtering is enforced [FS00]. See Figure 3.3 for description.

One must note, however, that this attack is applicable also for the popular sequential replacement approach.

3.2.1.2 Implementation issues

There are some practical issues regarding the real security of measurement traces. One problem is that the encryption key is kept in memory all the time and may be written on a swap disk. It would be possible to mark the memory location containing both the original key and expanded key non-swappable.

It can be considered, however, that the probability that the key gets written to

disk is quite low because:

- The keys are accessed frequently.
- The measurement system is a real-time one that should be dimensioned so that there is no need to use disk for virtual memory to guarantee acceptable performance.
- Physical security of equipment location must be maintained.

The selected key size (128 bits) is considered to be safe for now. In future, there may be a need to make the key longer. As an extra measure, a part of the secret is appended to the data to be encrypted. If there is some way to do brute force search, this should make it a bit harder.

3.3 Measurements for this work

The measurement period of 537 days started October 10, 2000 and ended by the end of March, 2002. A summary of the traffic statistics is shown in Table 3.1. Due to hardware problems, there are missing time periods from traces so data used in the analysis includes data of 431 days.

There are two sets of measurements, one including packets that arrive from hosts *external* ($\mathcal{E}$) to the laboratory and packets that depart from hosts *in* ($\mathcal{I}$) laboratory.

The user impatience study (Chapter 6) used the same datasets. For that study only TCP connections to port 80 were included. Summary of this subset is in Table 3.2.

Table 3.1: Statistics of measurements.

Category	$\mathcal{E}$	$\mathcal{I}$
Packets	600,945,166	503,318,744
of TCP	539,840,695 (89.8 %)	484,013,626 (96.1 %)
of UDP	56,847,171 (9.5 %)	16,168,059 (3,2 %)
Bytes	435,083,338,789	261,691,633,560
of TCP	394,736,887,275 (90.7 %)	257,054,366,316 (98.2 %)
of UDP	39,572,939,529 (9.1 %)	3,755,149,419 (1,4 %)
Source hosts	380,902	128

Table 3.2: Statistics of user impatience measurements.

Category	$\mathcal{E}$	$\mathcal{I}$
Connections	4,810,029	3,605,436
Aborted connections	621,720 (13 %)	977,791 (27 %)
Bytes in aborted connections	26.7 %	39.1 %
Unique client hosts (IP)	141,827	95
Average number of connections per host	25	41,826
Median number of connections per host	3	20,034
Average TCP throughput [bit/s]	127,275	246,045
Median TCP throughput [bit/s]	23,019	28,872
Average transfer size [B]	15,909	30,755
Median transfer size [B]	3,443	7,556

Chapter 4

Identifying network traffic classes

A large number of applications is being used in the Internet. Currently IANA lists more than 3,400 assigned TCP and UDP port numbers for different applications protocols.[1] A TCP or UDP port number would seem like an ideal choice for an application identification. There are, however, a few problems associated with this approach.

4.1 Problems of port-based identification

To start with, some applications do not use fixed port numbers at either end but select both ports dynamically. Examples of this type of protocols are RTP and ONC RPC, which negotiate port numbers before data connection is established.

Another problem is that common protocols are used in non-standard ports. While some protocols such as SMTP must use well-known ports, for other protocols it is possible to specify port used. There are at least two specified methods to select non-standard port: URLs and DNS SRV records. For URLs, some schemes allow specifying port number with host name, for example `<URL:http://www.example.com:123/>` would instruct web browser to connect to port 123 instead of default 80 on host `www.example.com`.

The DNS SRV records are not yet widely used but this may change in the future. They records allow locating services in a possible redundant way. If a client wants to access LDAP server for domain `example.com` using TCP, it first queries for SRV record for `_ldap._tcp.example.com`. In a successful case, it receives a list

[1] In most cases, both UDP and TCP port numbers are allocated for an application even if it uses only TCP or UDP.

of host names and port numbers with priorities and preference weights [GVE00].

Further, an application may use non-standard ports to masquerade as another application. Some protocols may be given higher priority based on port numbers in a network. One may benefit from using non-preferred application with the same port numbers as used by high-priority protocol. Another reason to masquerade is to go through a simple firewall that only filters by port numbers allowing only important or "useful" applications to pass.

If an application may be in different ports, it is also possible that an application may have different roles. An example of this kind of application is SSH. The SSH protocol supports terminal connections, file transfers and arbitrary TCP tunnelling and it uses the same port number for all of those – even in one TCP connection there may be multiple different connections. Similar problems exist with other tunnelling protocols.

To conclude, a foolproof identification of an application using only port numbers (i.e. stateless information) is difficult. However, each application has its own way to communicate. One can differentiate applications by comparing packet size and packet interarrival time distributions and flow lengths. This decision cannot, however, be made without state information or real-time, so one possible way to utilise this advanced identification is to verify performance of port- and address-based identification.

4.2 Application types and their characteristics

As noted before, there are a vast number of applications specified. In addition, each application protocol can be used in different ways, for example SSH as explained above. Still, it can be claimed that the majority of protocols can be classified into a few distinct classes based on their properties. In the following these classes are described.

4.2.1 Dialogue applications

Dialogue-type applications transfer data both ways, typically in turns. It depends on type of application how much data is transferred in each turn. It is also possible that both end points transfer at the same time if application in use supports pipelining or similar asynchronous execution.

4.2.1.1 Terminal applications

Remote terminal or virtual terminal is one of oldest applications used over a network. Instead of connecting terminal directly to a host computer, connection was carried over the network. A user types in commands and those are transported to the other host and responses are printed back.

The data flow is typically asymmetric: when the user types a character these are echoed back. Every now and then a larger fragment of data is received. Models of interactive telnet sessions can be found from [Pax94, PF95].

4.2.1.2 Command-response dialogue applications

Many Internet applications are line-oriented: a client connects to a server and issues one-line commands (some tens of characters) to which server replies by one or few line response. An example of this kind of application is FTP. The FTP protocol uses separate TCP connection to transfer files [PR85].

4.2.1.3 Command-response with embedded transfers

This application is quite similar to the previous one. The difference is that also longer data fragments are transferred using same connection thus there are short packets and also long packets. Examples of this kind of application are NNTP, IMAP and POP message transfer protocols [KL86, Cri96, MR96].

The SMTP protocol is different in the sense that bulk transfer goes towards the server while in other protocols the direction of data transfer is from the server to the client. The NNTP protocol is bi-directional in data transfers as a client may also send articles. In NNTP server-server communications articles are exchanged both ways.

An example of SMTP dialogue is shown in Figure 4.1. The server indicated in response to `EHLO` that it supports pipelining. The client can send both the sender (`MAIL FROM`) and recipient (`RCPT TO`) commands in one burst. Use of PUSH flag indicates end of command or end of document transfer.

Time		⇒	Flags	Bytes	Comment
0.000	$\mathcal{C}$	$\mathcal{S}$	S	0	Client initiates TCP connection
0.785	S	$\mathcal{C}$	$\mathcal{S}$	0	Server responds
0.786	$\mathcal{C}$	$\mathcal{S}$	.	0	Handshake completed
1.835	$\mathcal{S}$	$\mathcal{C}$	P	49	Initial status code
1.835	$\mathcal{C}$	$\mathcal{S}$	.	0	Acknowledged (TCP)
1.836	$\mathcal{C}$	$\mathcal{S}$	P	18	Client sends EHLO
2.725	$\mathcal{S}$	$\mathcal{C}$	.	0	
2.855	$\mathcal{S}$	$\mathcal{C}$	P	271	Server reports capacities
2.856	$\mathcal{C}$	$\mathcal{S}$	P	80	Sender and receipient
3.705	$\mathcal{S}$	$\mathcal{C}$	P	35	“Sender ok”
3.715	$\mathcal{S}$	$\mathcal{C}$	P	32	“Receipient ok”
3.735	$\mathcal{S}$	$\mathcal{C}$	P	40	“Send data”
3.736	$\mathcal{C}$	$\mathcal{S}$	.	0	
3.736	$\mathcal{C}$	$\mathcal{S}$	.	1364	Email body
3.736	$\mathcal{C}$	$\mathcal{S}$	.	1364	. . . continues
5.545	$\mathcal{S}$	$\mathcal{C}$	.	0	
5.545	$\mathcal{C}$	$\mathcal{S}$	.	1364	. . . continues
5.546	$\mathcal{C}$	$\mathcal{S}$	P	159	. . . continues
6.826	$\mathcal{S}$	$\mathcal{C}$	.	0	
7.826	$\mathcal{S}$	$\mathcal{C}$	.	0	
7.836	$\mathcal{S}$	$\mathcal{C}$	P	15	“Received ok”
7.876	$\mathcal{C}$	$\mathcal{S}$	.	0	
7.886	$\mathcal{C}$	$\mathcal{S}$	P	6	“Quit”
7.886	$\mathcal{C}$	$\mathcal{S}$	F	0	
8.756	$\mathcal{S}$	$\mathcal{C}$	P	34	“Server closing”
8.756	$\mathcal{S}$	$\mathcal{C}$	F	0	

Figure 4.1: SMTP dialogue as network trace. Packet exchange between the client ($\mathcal{C}$) and the server ($\mathcal{S}$) are listed one by line. “Flags” column indicates TCP flags and “Bytes” are application bytes excluding IP and TCP header.

4.2.2 Transaction applications

4.2.2.1 Short command-response with no persistence

One of most widely used transaction applications in the Internet is DNS which takes care of mapping names to IP addresses, locating services and similar overlay network management. In a typical scenario, a client sends *recursive* query to a local DNS server which then performs zero or more *non-recursive* queries to find the answer.

If a name server receives a recursive query, it tries to resolve an answer to the query and return the answer to the client. If the query is non-recursive, the server replies with an answer if it knows the answer or with a pointer to another server, which should know more about query.

There are different, non-exclusive roles a name server can have. The first role is to act as an authorative name server for a domain. The server receives queries from different servers all over the network and there is not much of persistence in flows. Another role is that of a resolving name server that collects necessary information on behalf of its clients. It receives a number of queries from a small local set of hosts and queries servers all over the network.

4.2.2.2 Command-response with long data part

Probably the most common application protocol prevalent in the Internet is HTTP, which is used for web document transfers. The first widely used version of the protocol (1.0 [BLFF96]) was a simple one. A client sent a request to a server, which replied with a response (Figure 4.2). The request included the operation, typically `GET` or `POST`, request-URI and HTTP version as the first line and possible header lines that provide additional information about the request and the client. The median request size was approximately 300 bytes excluding IP and TCP headers for measurements described in Table 3.2. More than 95 % of requests were less than 1000 bytes in size.

The server replies with a response that includes header fields and the document itself. The header fields provide document and server information, such as a last modification time and a media type of the document. The median response size was 7,500 bytes for the dataset $\mathcal{I}$ and 3,500 bytes for the dataset $\mathcal{E}$ with 95 % quantile at 20 and 50 kibibytes, respectively.

Time		⇒	Flags	Bytes	Comment
0.000	$\mathcal{C}$	$\mathcal{S}$	S	0	Client initiates TCP connection
0.829	$\mathcal{S}$	$\mathcal{C}$	S	0	Server responds
0.830	$\mathcal{C}$	$\mathcal{S}$	.	0	Handshake completed
0.830	$\mathcal{C}$	$\mathcal{S}$	P	388	Client sends request
2.170	$\mathcal{S}$	$\mathcal{C}$	.	512	Server responds
2.170	$\mathcal{C}$	$\mathcal{S}$	.	0	
2.300	$\mathcal{S}$	$\mathcal{C}$	.	512	... continues
2.300	$\mathcal{C}$	$\mathcal{S}$	.	0	
3.080	$\mathcal{S}$	$\mathcal{C}$	.	512	... continues
			Document is transferred		
4.930	$\mathcal{S}$	$\mathcal{C}$	FP	112	Last segment, closed
4.930	$\mathcal{C}$	$\mathcal{S}$	F	0	Client closes
5.570	$\mathcal{S}$	$\mathcal{C}$	.	0	Final ACK

Figure 4.2: HTTP dialogue as network trace.

Version 1.1 introduced persistent connections making it possible to perform multiple request-response exchanges over one TCP connection [FGM+99]. This improves network performance as TCP can better adapt on network conditions and latency in connection establishment is avoided [NGBS+97]. It is also possible to pipeline requests further improving performance if the requests are non-idempotent, i.e. do not have side effects if failed [FGM+99, p. 46].

The use of persistent connections changes flow properties somewhat by removing strict alternation from client-server pair. Based on the measurements, multiple document transfers constitute still a small proportion of all transfers. More than 90 % of HTTP connections had only one request and less than one percent had four or more. Some of the connections had tens of requests.

4.2.2.3 One-way transfers

FTP (see Section 4.2.1.2 above) uses TCP connections to transfer files. One file is transferred over one TCP connection. This transfer is one-way. On the network, large segments travel in one direction and acknowledgement-only packets travel in another direction.

4.2.3 Streaming applications

Streaming applications are the ones which transmit real-time or near real-time media content, typically audio and video. So far, many applications are based on proprietary technologies and there exists little documentation on those. According

to [MH00], the Real audio flows over UDP have a near-constant packet size that depends on the media target rate. Packets were not sent at constant intervals but they were sent in small bursts.

There were only few instances of streaming applications that could be reliably identified. According to [LIP99], where streaming media and VoIP traffic were monitored, real-time applications have both their packet size distribution and packet interarrival time distribution uni-modal.

4.2.4 Network scans

Malicious network users try to locate systems that are easy to compromise, most often to use them as stepping-stones to hide their true origin in attacks against other systems. The network scans can be divided into two main categories: those trying to find out services available at a particular host and those trying to find out if there is any host that provides some service. The service in question may be one with a security vulnerability or a trojan software.

These two types both result a large number of short flows. In the first case the destination port number changes and in the second case the destination address changes. One of the identifying factors for the latter case is that there is traffic to non-existent hosts in the network. A malicious worm, such as Code Red, may initiate network scans to locate vulnerable servers it could infect.

For purpose of this study, only simple scans were studied. From a 48-hour, 5-tuple $\mathcal{E}$ dataset TCP flows with only one packet were selected. Furthermore, all packets destined to port 113, identification protocol [Joh93], were ignored as only few hosts run identification protocol server which could reply to queries. Packets are filtered based on their destination port at HUT access router. The filtering applies on Windows networking protocols and SMTP among other and reduces proportion of those protocols if not completely removes. Protocols are further filtered at Networking laboratory access router. The measurement location is between access routers, thus end systems do not see every packet the measurement system records.

There was 361,270 probes from 23,995 source addresses.[2] Daily count over measurement period is shown in Figure 4.3 with monthly average. One should note that also licit attempts might record as network scans if the server fails to respond. However, a client usually resends connection establishment packets a few

[2]Because of possible source address spoofing, we cannot know the exact number of *hosts*.

times and the flow will have more than one packet.

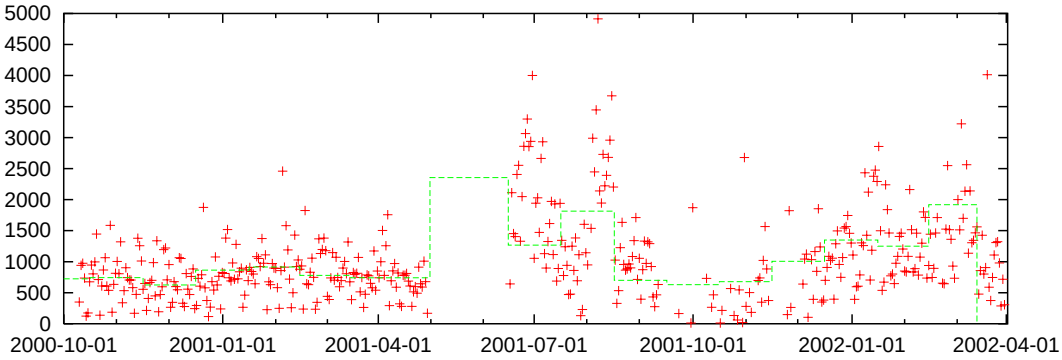


Figure 4.3: Daily count of network scan packets. Outside of figure is 2002-03-03 with 19,479 packets.

The most popular destination ports were 21 (FTP), which received 44,532 probes, 22 (SSH, 21,223), 80 (HTTP, 16,448), and 53 (DNS, 12,716). All of those services have security vulnerabilities in some widely used implementation or rife misconfiguration. Based on sample of source addresses, a significant portion of scans originates from home computers with broadband access, small business and schools; from organisations that do not always have a professional computer administration. Those systems are probably compromised and used to locate more vulnerable hosts.

Another group of ports are services that are not accessed over Internet but are for local use only. These include ports 515 (printer, 6,420), 111 (portmapper, 3,678), 6,112 (dtspcd, 2,241), and 98 (linuxconf, 425), which all have vulnerabilities. In general, there is not licit use for those services outside of organisation.

Third group includes scans for backdoors opened by exploits on vulnerable services or by trojan software. Those backdoors listen on certain ports and do not have any access control by default. One can search for backdoors opened by email viruses, trojans, worms or exploits even if one has not initially broken in. For example, the Code Red II worm made it possible to execute any commands on infected host using HTTP commands [SPW]. Popular backdoors included several ports used by Sub Seven trojan, Senna Spy and backdoor by statd exploit [CER99].

4.2.4.1 Life span of whole-network scans

Hosts, which scanned more than 100 hosts and sent only one packet to each host, were selected from above set resulting 285 hosts. Time from the first packet to

the last packet was measured for each source address. The fastest one sent 254 probes in 0.027 seconds while the slowest one used 2 hours 20 minutes for 212 probes. The median was 36 seconds. A cumulative plot is in Figure 4.4.

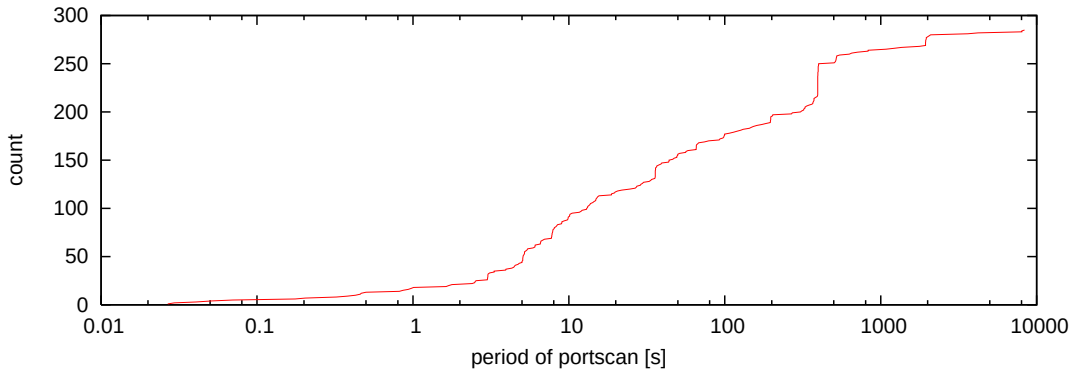


Figure 4.4: Cumulative count of one-pass full scans

4.3 Conclusions

The applications used in a network are dissimilar. Each application has its own logic and that will result in network traffic typical to the application in question. Based on these characteristics it would be possible to identify applications in the network.

Network scans are a different class of "applications". Identifying those can help protecting network from break-in attempts. While it would be easy to notice scans with packet rate of 9,000 packets per second, slower ones are difficult to identify, especially if done in distributed manner [SPW].

Over time, applications have been used as substrate to another application if the other application is not available [Moo02]. For example, if a host did not provide FTP service [PR85], files could be transferred using modem transfer protocols over telnet connection [PR83], which is a very inefficient method. A modern equivalent is (ab)use of HTTP to pass through firewalls. Even streaming media is being delivered over HTTP connections.

Chapter 5

Flows of network traffic

5.1 Flow definition

A flow is a series of packets travelling from one part of the network to another [JR96]. A flow can be either *unidirectional*, i.e. packets travelling from A to B belong to a different flow than those travelling from B to A, while in a *bi-directional* flow they belong into the same flow. While a bi-directional flow data provides better insight into the behaviour of individual protocols and applications, it is not always readily available in the core network because of asymmetric routing [CBP95, Pax96].

5.1.1 Flow granularity

A flow can be defined with very different granularities. In a core network, the operator is usually interested in how traffic flows from one AS to another. To provide a QoS class for an application, a transport layer port numbers must be used. Possible granularities can be defined as in [Lof97], with extensions by the present author, as follows:

- application, identified by
  - TCP or UDP port numbers
  - transport protocol
  - IPSec SPI [BO97]
  - IPv6 flow identifier

- host, identified by
  - network layer address (IP address)
  - link layer address (e.g. MAC address)
  - hostname (e.g. DNS name)
- network, identified by
  - address prefix
  - AS number
  - domain name
  - arbitrary group of hosts
- traffic sharing a common path in the network, identified by
  - link (interface on router)
  - ATM or FR virtual channel identifier
  - MPLS path
  - AS path

Different granularities are useful on different occasions. It is possible to define additional granularities based on other criteria, but to be useful for classification, for example, the decision must be made on information contained in a single packet.

Different granularities need different amount of information, but a typical case needs following information for IPv4:

- source host IP address,
- destination host IP address,
- transport protocol,
- source port,
- destination port,
- start time of flow, and
- end time of flow.

This information is enough to identify a flow up to the transport layer. If there exists some multiplexing on the application level, e.g. when using SSH or some other tunnelling, it cannot be identified. For example, to accurately identify properties of documents transferred over HTTP connection, one must record all of TCP segments and compile all of the transferred streams [Fel98].

An interesting variant of a flow definition ignores the destination port number. Using this definition it is possible to join all communication between a client and server using same protocol to one flow. This is useful as the client port number is random and possibly changes for every (TCP) connection.

5.1.2 Timeout for flow

As *a flow is a series of packets travelling from one part of network to another part of network* it is also limited in time. Packets are aggregated over time period and the ones sharing some identifier for a flow, belong to the same flow. A flow is considered active as long as the inter-packet intervals are shorter than selected timeout value.

The timeout value varies depending on analysis requirements and available processing and storage capacity. Based on the study in [CBP95], a timeout value of 64 seconds (or 60 seconds) is commonly used. Other values used are 600 seconds in NeTraMet monitor [NeTb] and 1800 seconds (30 minutes) in Cisco NetFlow[1] [Neta].

Another issue is the capture time used. Because of vast memory requirements, captures are usually 15-minute or one-hour samples taken from the network. While most of flows are well shorter than one hour, there exists considerable number of flows that are longer. This is studied in Section 5.2.1.

These flow timeouts do not catch longer periodic cycles of network traffic. To capture natural cycles of day and week a longer timeout is needed. If a news web site, which a user visits daily, is observed, there is a persistent flow which has interarrival times of approximately 24 hours between daily visits, interarrival times of few minutes as user reads pages, and sub-second interarrivals to retrieve page components.

Weekly behaviour can be captured if the flow timeout is set to 48 hours. This will result in a flow being broken over weekends and vacations but persisting over

[1]Netflow uses shorter timeout value if there is not enough memory to store default period.

working days. Also destination port should be ignored if one wants to include all of client-server traffic into a single flow.

5.2 Results from flow measurements

For this work, three different definitions of flow were used.

1. Complete 5-tuple was used for TCP and UDP. For other protocols (like ICMP) source host, destination host and protocol defined a flow.
2. For HTTP impatience study (Chapter 6) flows were delimited by SYN/FIN or SYN/RST segment pairs.
3. The destination port was ignored from 5-tuple and flow timeout was 48 hours for all protocols as it was not possible to follow TCP connection setup and sequence numbers when the destination port is ignored.

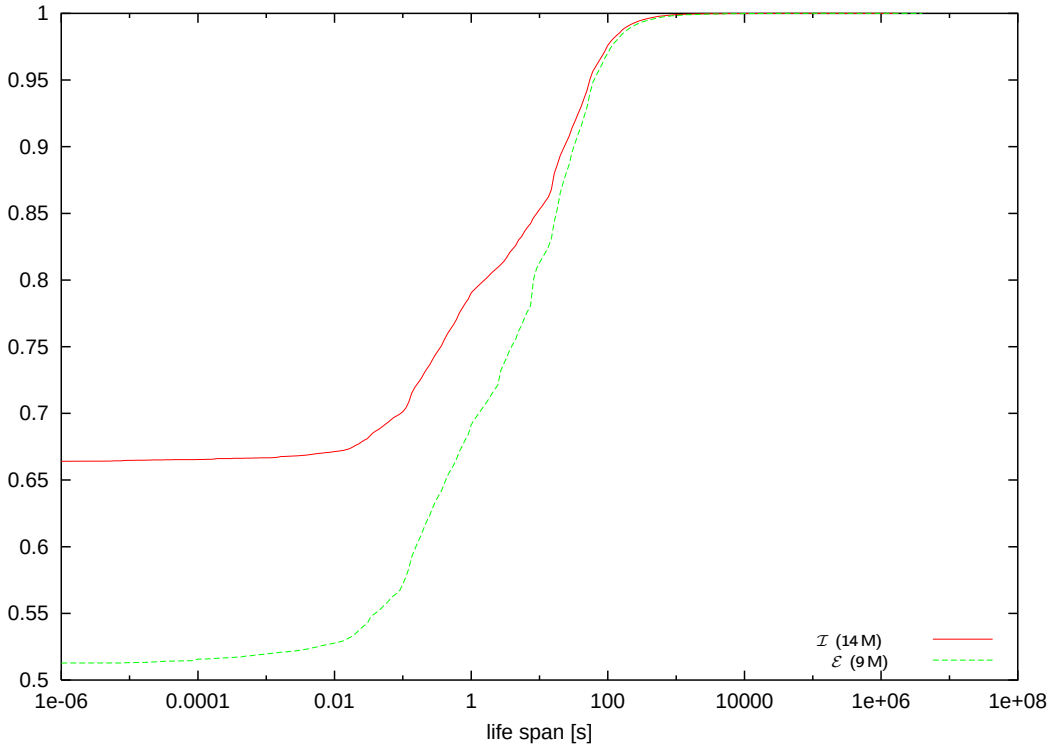
The latter one corresponds to a client accessing a service at a same server. This would reveal more of human behaviour using application or applications.

The user impatience study (Chapter 6) allowed a comparison between TCP connections and TCP flows for HTTP protocol. It was found out that a 60-second timeout with 5-tuple resulted in 3 – 7 % more flows than there were TCP connections. The figures were opposite if partial TCP connections were included. A partial TCP connection is one that is not ended with FIN or RST. A typical one is a connection attempt where a server fails to respond.

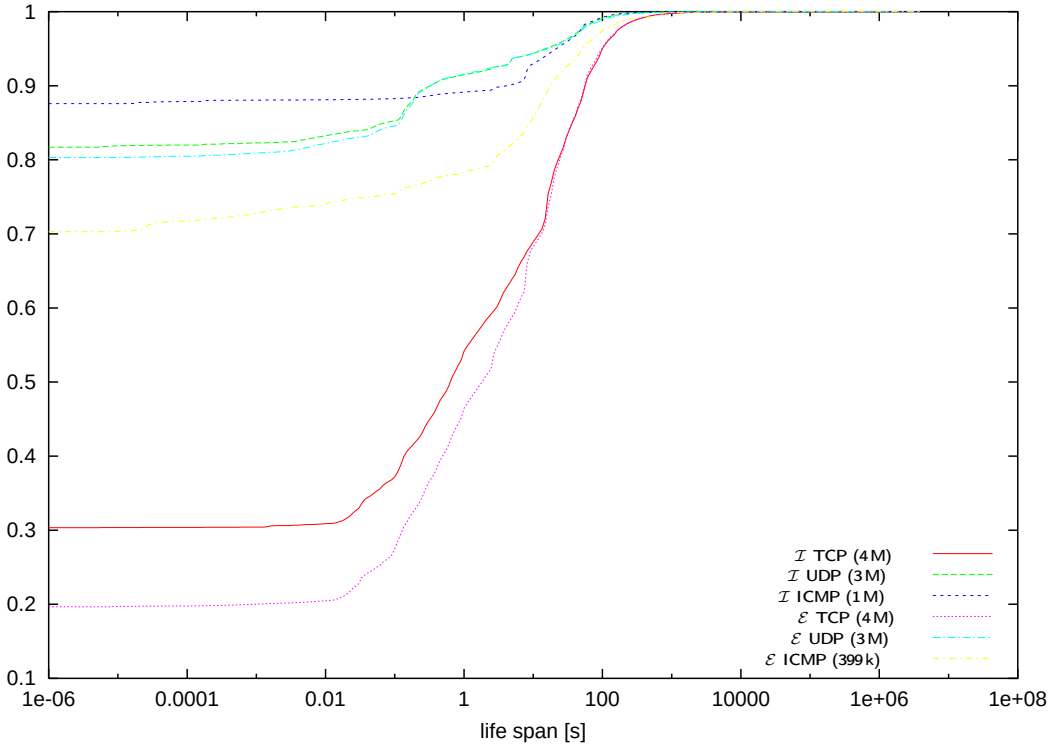
5.2.1 Distribution of flow lengths

An extended measurement period shows that many flows are very long and a great amount of bytes is transported over these long flows. If normal 5-tuple (see above) is considered, approximately 99,9 % of flows are shorter than one hour but they carry only 70 – 80 % of bytes.

Cumulative distribution of flow lifetimes and flow byte counts are shown in Figure 5.2 and 5.3.

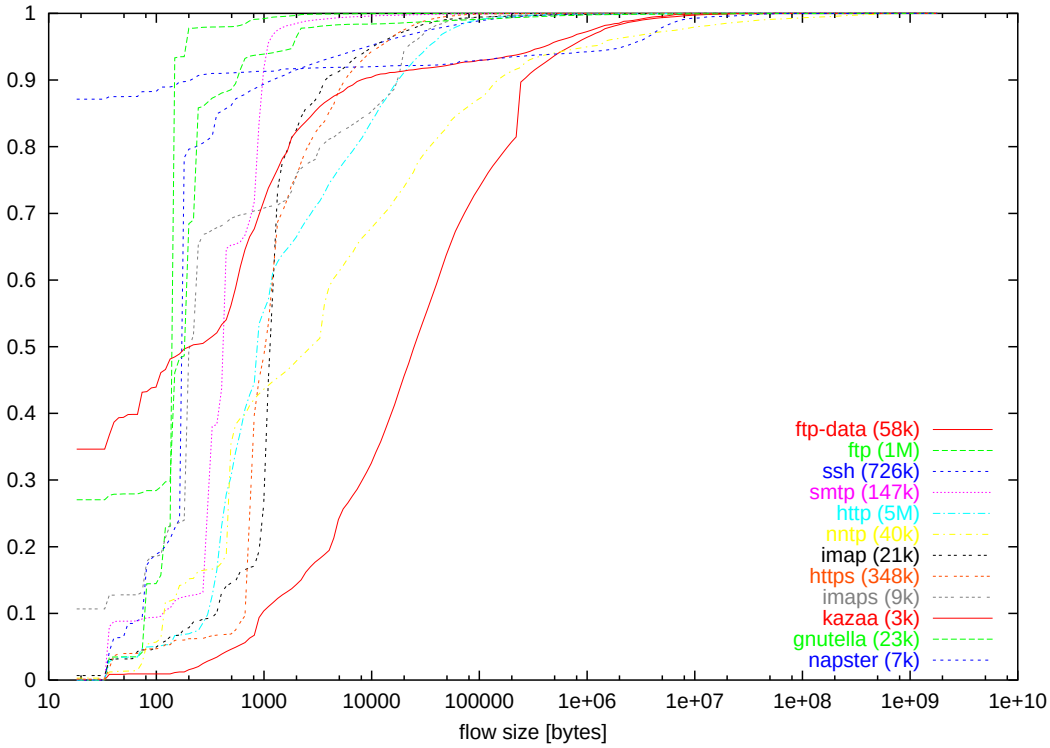


(a) (source, destination)

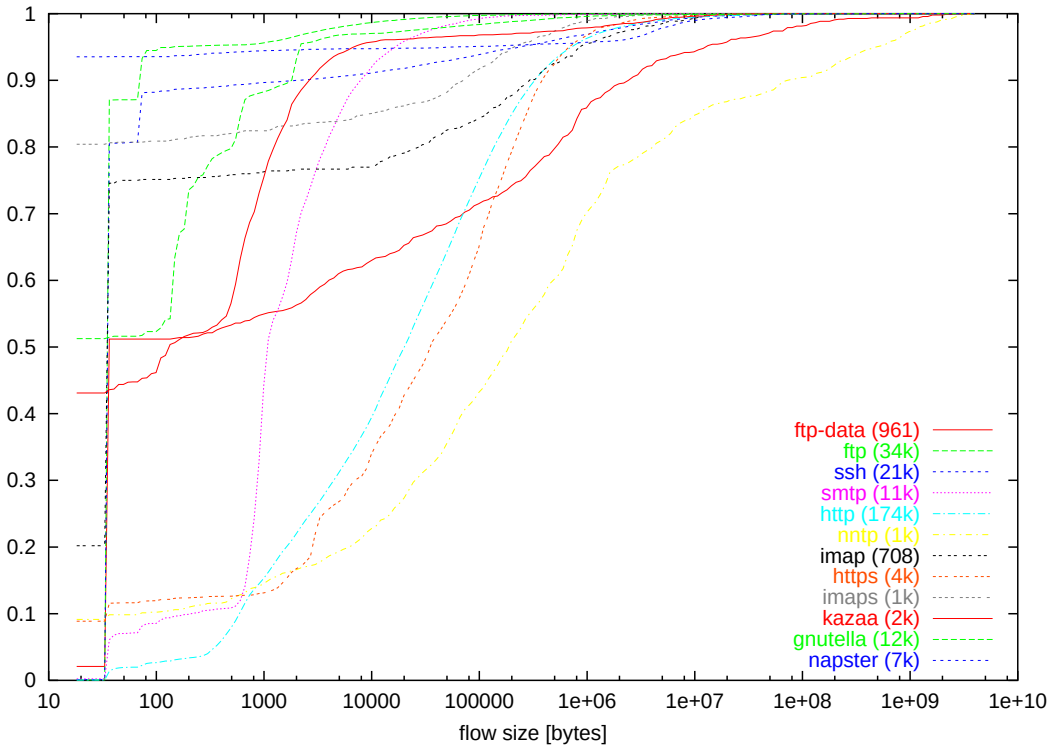


(b) (source, destination, protocol)

Figure 5.1: Cumulative distribution of flow lifetime with 60-second timeout.

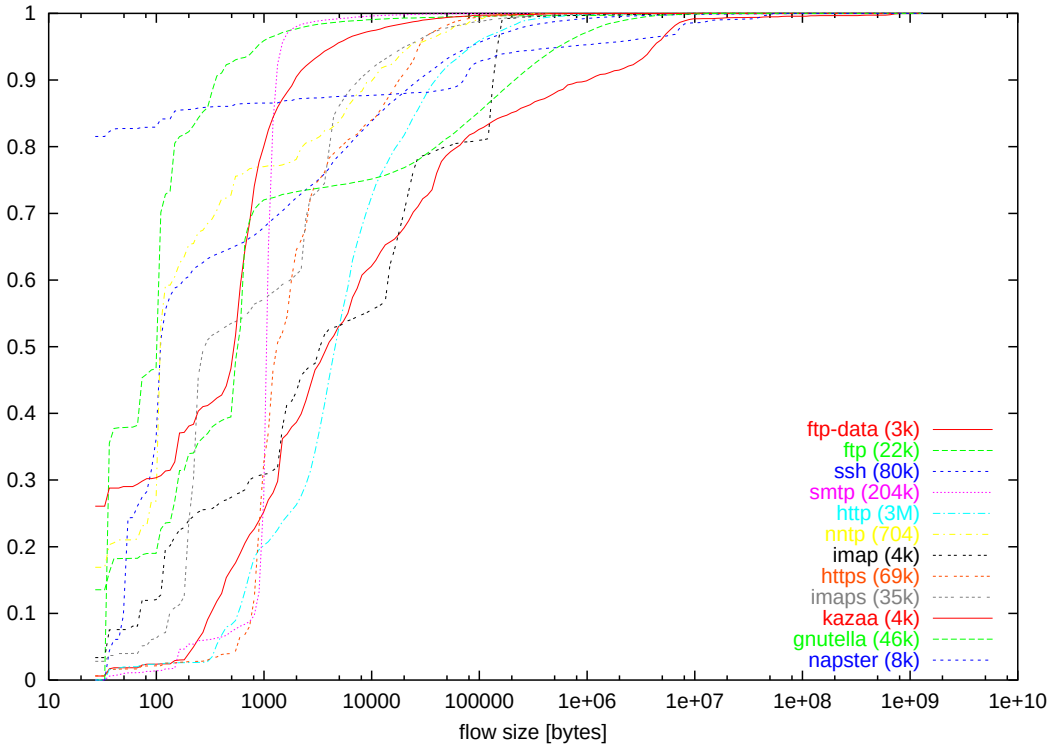


(a) 60-second timeout, 5-tuple

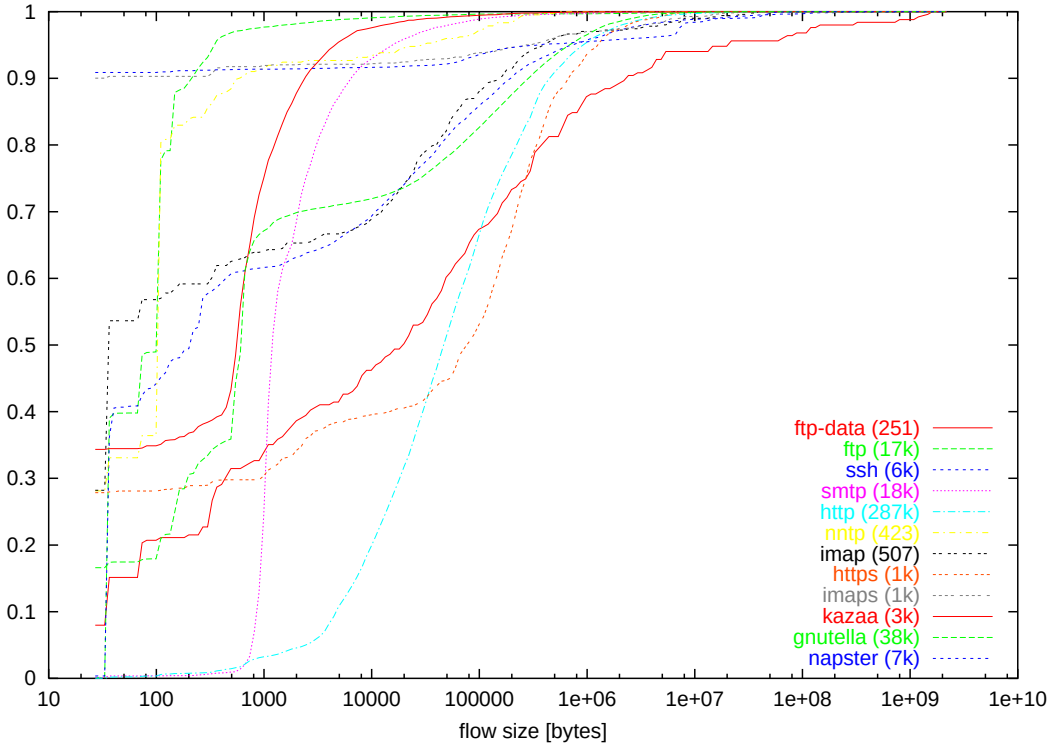


(b) 48-hour timeout, 4-tuple

Figure 5.2: Cumulative distribution of TCP flow lifetime for dataset $\mathcal{E}$.



(a) 60-second timeout, 5-tuple



(b) 48-hour timeout, 4-tuple

Figure 5.3: Cumulative distribution of TCP flow size for dataset $\mathcal{I}$.

5.2.2 Packet interarrival times by application

A group of most-used applications was selected for a more detailed study. The protocols include

- FTP, both command and data connections (ports 20 and 21),
- SSH (port 22),
- SMTP (port 25),
- HTTP (port 80), also secure HTTPS (port 443),
- NNTP (port 119), and
- IMAP (port 143), also secure IMAPS (port 993).

The client port number was ignored and flow timeout of 48 hours was used. The interarrival time for packet n was counted into bin N if the time difference $t_n - t_{n-1}$ was equal to or greater than $2^N 10^{-6}$ and smaller than $2^{N+1} 10^{-6}$ seconds.

5.2.2.1 FTP interarrival times

The file transfer protocol (FTP) uses two ports: one for commands (21) and another (20) for file transfers. Each file is transferred with one TCP connection. The command connection has typical command-response dialogue application and has potentially long intervals as connection is idle when file is being transferred.

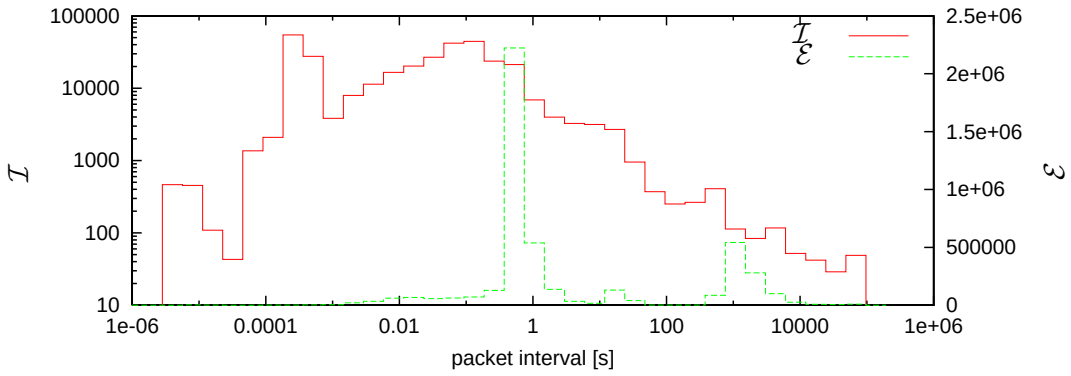


Figure 5.4: FTP command packet interarrival times.

The command channel has approximately 30 times more connections with the 5-tuple, 60-second criterion than with the 4-tuple, 48-hour timeout.

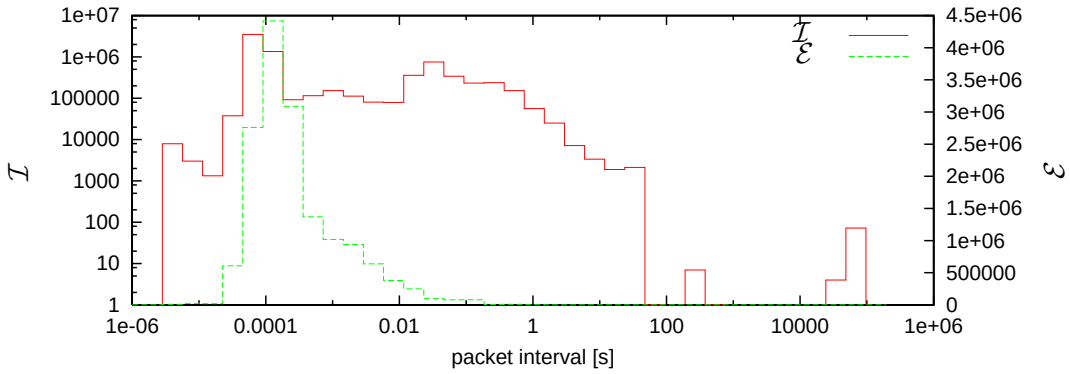


Figure 5.5: FTP data packet interarrival times.

5.2.2.2 SSH interarrival times

The secure shell is used for interactive terminal connections and to copy files over the network from one host to another. It is also possible to tunnel other TCP connections such as X11, POP, and SMTP over this secure connection. This superposition results in a mixture of different processes.

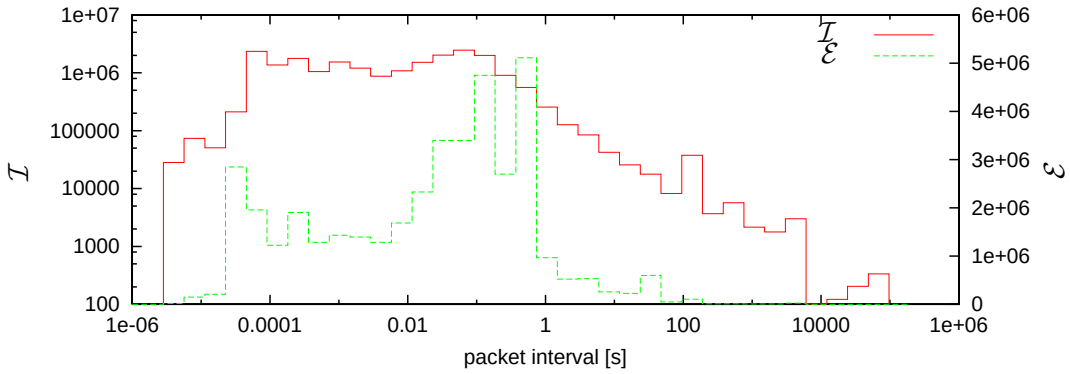


Figure 5.6: SSH packet interarrival times.

There is not much of difference (maximum twofold) in flow counts if 5-tuple or 4-tuple is used as criterion. Shorter timeout results in 5 to 20 times more flows than the longer one.

5.2.2.3 SMTP interarrival times

The SMTP protocol differs from most other protocols in the sense that a majority of data flow is towards the server, not from the server. As MTA has mail to deliver, it connects to a recipient SMTP server and after dialogue transmits data.

In Figure 5.7 it is possible to identify three regions of interarrival times. Times around 0.01 second correspond to acknowledgements of mail transfer. Times

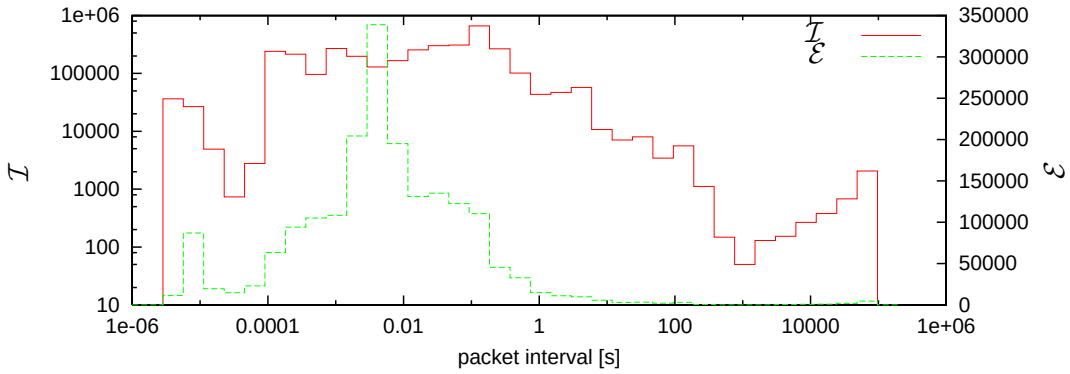


Figure 5.7: SMTP packet interarrival times.

around few seconds range result from server checking information (such as blacklists for unsolicited bulk email) or doing longer processing. Times over few minutes to few hours result from email users sending email messages.

For longer timeout values, the 5-tuple criterion has ten times the 4-tuple flow count. In shorter time periods there is not much of difference and the timeout value does not have much effect in the 5-tuple case.

5.2.2.4 HTTP interarrival times

The HTTP protocol contributes over 61 % of outbound TCP traffic. It is a transaction type protocol where a client sends a request and the server replies with response. It is possible to issue several requests over the same TCP connection.

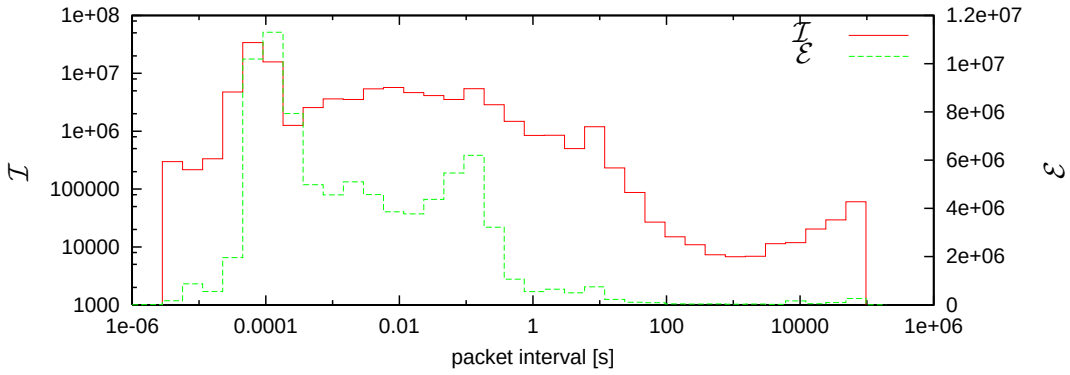


Figure 5.8: HTTP packet interarrival times.

For internal clients, there are quite a few connections at timescales from two hours to 20 hours. This matches to "visit few times at working days" profile.

5.2.2.5 IMAP interarrival times

The IMAP protocol is used for connections between MUA and MTA. The MUA is usually configured to check email every now and then, typical values are from 5 to 10 minutes. This can be seen from Figure 5.9 where there is a peak around 10 minutes.

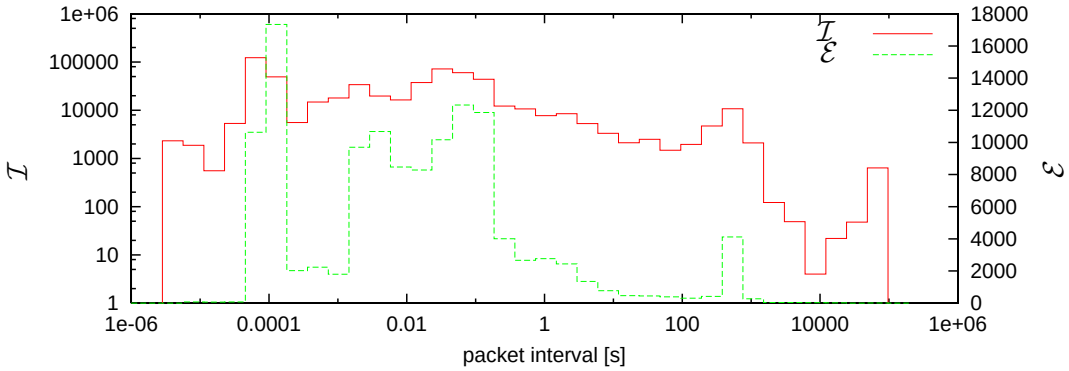


Figure 5.9: IMAPS packet interarrival times.

5.2.2.6 NNTP interarrival times

The Usenet news, which are accessed via the NNTP protocol, provide discussion groups on various topics. When a user opens a newsreader, it reads a list of article counts in newsgroups. Then the user reads articles, which are retrieved from the server one by one. As each article takes different time to read, the packet interarrival times are between a second and few minutes. There are only few connections from outside of the laboratory as the laboratory internal NNTP server (external clients) provides only small set of mailing lists archived to the news server.

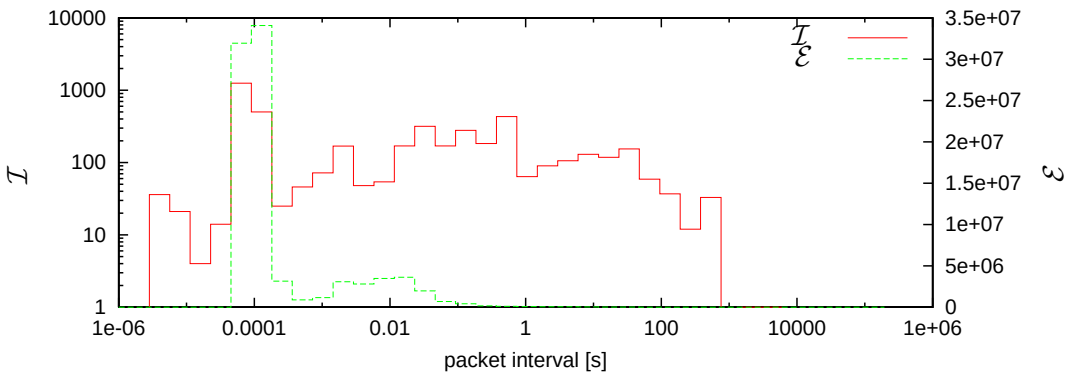


Figure 5.10: NNTP packet interarrival times.

5.2.3 Flow timeout relation to flow count

Each flow application has its own characteristic packet interarrival times as described in Section 5.1.2. Packet interarrival histograms were calculated over time periods from 1 µs up to 48 hours using histogram bin sizes of 2^N µs, $1 \leq N \leq 36$, i.e. every bin size twice as large as the previous one.

Overall distribution can be seen from Figure 5.11 for both flow resolutions (5-tuple, 4-tuple), both directions ($\mathcal{E}$, $\mathcal{I}$) and for TCP and UDP protocols.

There is a significant increase in flow count for UDP connections for a timeout around one minute. The NTP protocol for clock synchronisation queries a server once a minute for the correct time. This contributes most of the increase as can be seen from Figure 5.12, which includes a set of the most popular application protocols on top of UDP.

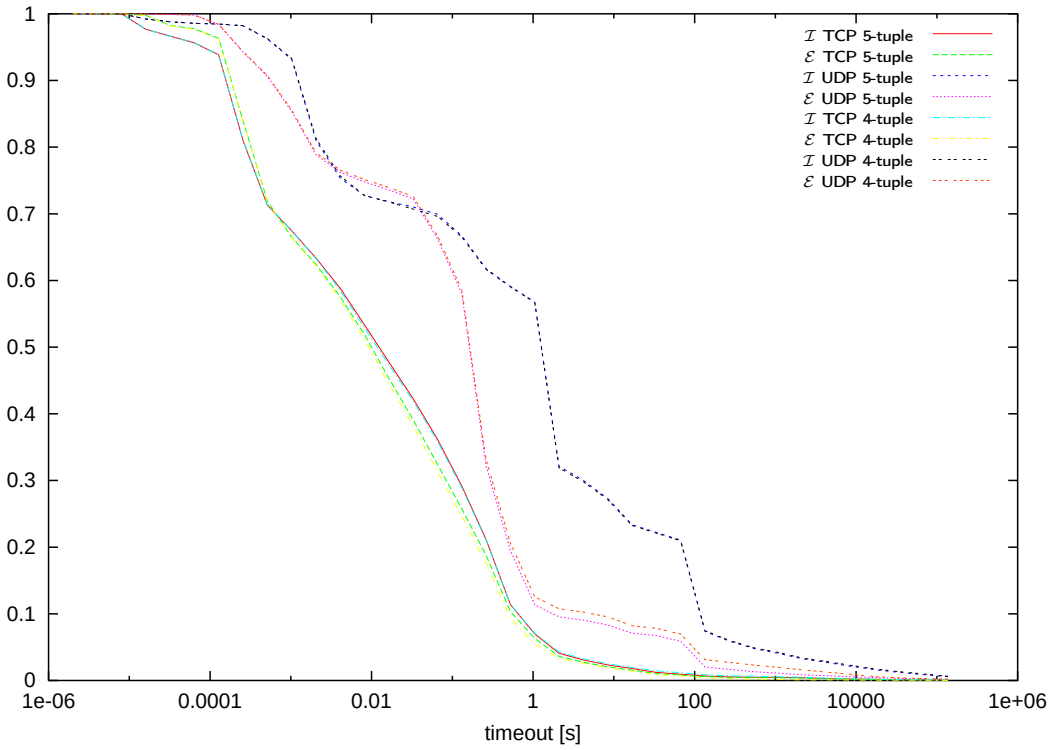
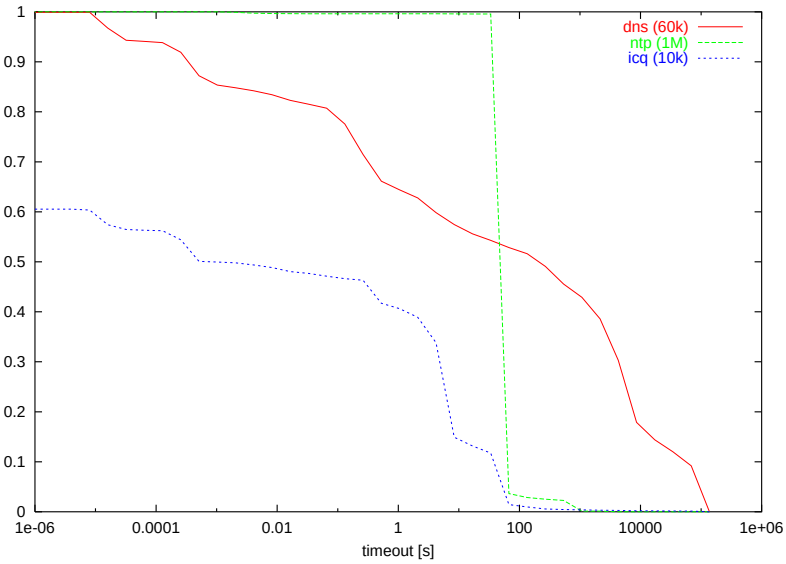
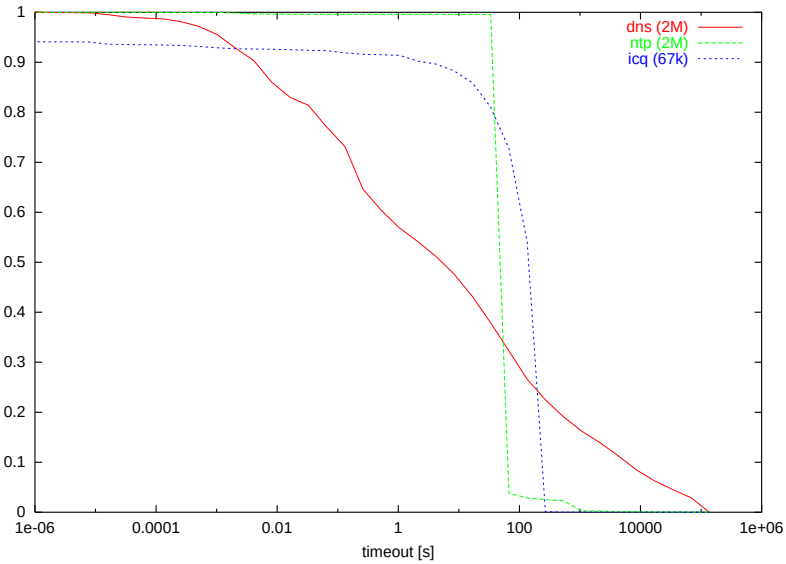


Figure 5.11: Flow count as function of timeout.

Overall distribution is shown in Figure 5.11 for both flow resolutions (5-tuple, 4-tuple), both directions ($\mathcal{E}$, $\mathcal{I}$) and for TCP and UDP protocols.

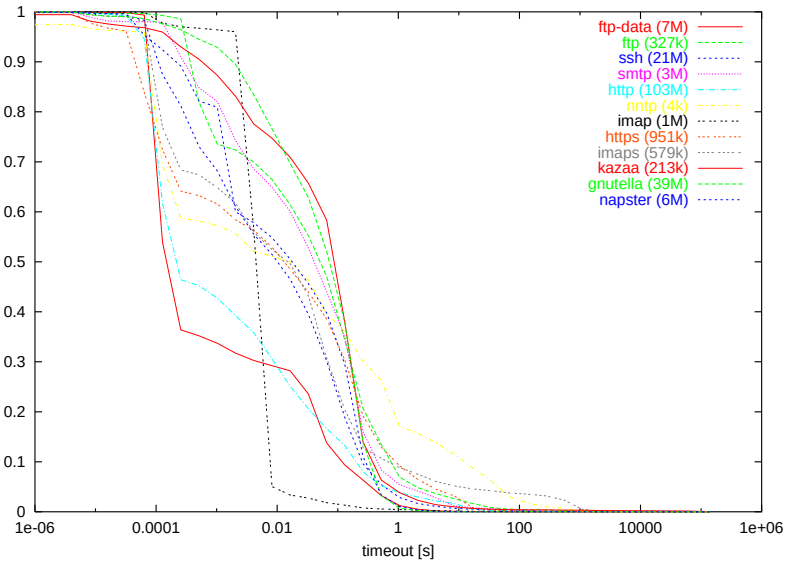


(a) $\mathcal{I}$

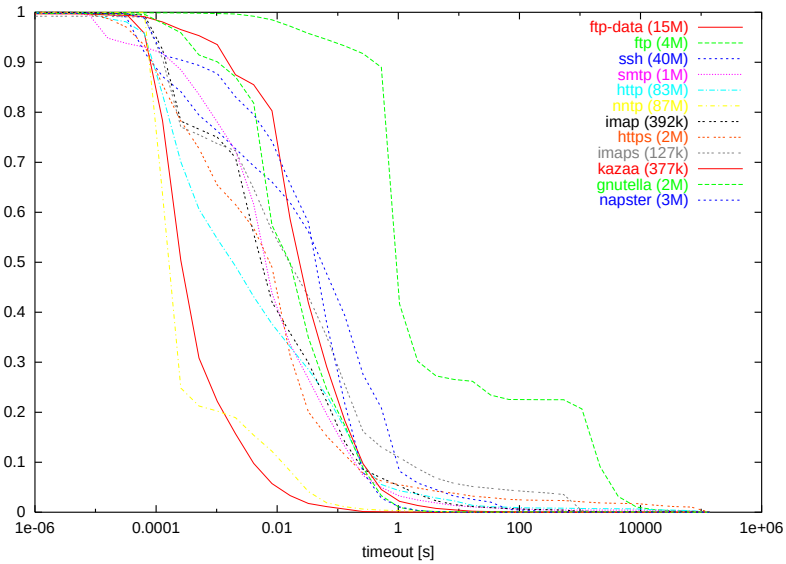


(b) $\mathcal{E}$

Figure 5.12: Flow count as function of timeout for set of UDP protocols.



(a) $\mathcal{I}$



(b) $\mathcal{E}$

Figure 5.13: Flow count as function of timeout for set of TCP protocols.

5.3 Conclusions

The anticipated results based on analysis in Chapter 4 were found. Different applications do have different flow properties. Packet interarrival times reflect the underlying application logic and, for interactive applications, user behaviour.

Some applications are very sensitive for both the granularity and timeout, some like HTTPS only for granularity. If wants to account network traffic at flow level, using a 4-tuple would reduce the count of flows to half of the flow count using a 5-tuple.

Chapter 6

User impatience

User impatience is an essential factor affecting network utilisation. It sets the limit for the time a user is willing to wait without one's task proceeding or any indication of that. The network should fulfil the interactive user's request to transfer data as soon as possible. This results in non-smooth traffic demand.

Impatient users pose presently a problem for the telephone network operator. In a telephone network, call setup occupies processor resources in telephone exchanges and reserves capacity from intermediate links. If the caller aborts the connection before the callee answers, the operator gets no revenue from the call. If the caller thinks that the call does not proceed because of the network, the unsuccessful call also results in dissatisfaction on the operator. The network should be designed so that the PDD, the post dialling delay before a ringing tone is received after dialling the last digit [E.492], does not exceed the time most people are willing to wait for a ringing tone.

A similar scenario can be found in packet networks like the Internet. If a user tries to access a web page, it results to several packets travelling in the network. Depending on how the page is designed, there may be several connections to different servers, each with different quality. If the user gets impatient and aborts the transfer before the page is completely loaded, the page may or may not be useful. If the page is not useful for the user, all data transferred have used network resources in vain.

An unsatisfied user is a concern to the operator, but depending on the charging scheme – whether flat rate or byte-based – the unnecessary data may be a problem for the operator also as it takes resources from other connections.

User impatience in the telephone network has been studied for a long time. The

delays in a telephone network are mainly dependent on the number of telephone exchanges the call traverses through. In IN services, such as FreePhone, delays can be more variable as there is communication between different IN entities, database lookups and variable processing delays [Jor94]. In cellular networks, one must balance between reneging of new customers and terminating calls on hand-off [CCL99]. The author is not aware of any web traffic related studies about user impatience.

The delay between a user action and the response is one of the important usability factors. Usability studies have shown that if a user gets feedback for his action within 0.1 second, the response is fast enough. If the task completes within 1 second, the user's workflow does not get interrupted. If the task does not complete within ten seconds, user attention is lost [Nie93, p. 135].

The Web use does not meet these requirements. Many web pages have not been designed with the speed in mind. Even if the Internet is not congested, pages take well over ten seconds to download for a typical PSTN modem user [Nie97]. A user may have different expectations based on connection one is using. The ITU E-model [G.100] is a computational model in transmission planning to derive transmission rating factor R. To project user expectations in different environments, it includes an *advantage factor* A that ranges from 0 on "normal" telephone to 20 on multi-hop satellite connections or similar hard-to-reach locations. In Equation (6.1) R_o is a basic signal-to-noise ratio. Impairment factors are divided to simultaneous factors (I_s), delay factors (I_d), and equipment factors (I_e).

$$R = R_o - I_s - I_d - I_e + A \tag{6.1}$$

Values of $R < 0$ correspond mean opinion score (MOS) of 1 and values $R > 100$ MOS of 4.5. Values between are interpolated using cubic function. The maximum value of A is equivalent to one MOS step, i.e. the same quality results "poor" (2) on "normal" telephone and "fair" (3) if user expectations are very low.

In Section 6.1 the user impatience is studied, where the delays originate from, and how one can identify impatient users in the network. Section 6.3 contains the results from the measurements. Findings are concluded in Section 6.4.

6.1 User Impatience

An impatient user can be seen as a customer to a queueing system, who departs before being completely served: he may depart before the service has begun or during the service. The most commonly used model for the impatience is the memoryless model: in each small time interval Δt the probability that a surviving connection will be aborted is constant.

6.1.1 Delay for the User

The delay in accessing web pages can be divided into several components. The user experiences the total transfer time for all components on a page. A client requests each element on the page with an HTTP transaction from the server. If both the server and client support HTTP/1.1, multiple elements from the same server can be transferred over one TCP connection.

In each TCP connection several TCP segments are transmitted. Each segment is subject to dropping or delay at some point in the network because of congestion. The total delay for each segment has two components:

- Deterministic delay, which is caused by the physical propagation delay in the transmission medium and the time the network elements need to process the packet including the transmission delay. This can be referred to as the "delay in an unloaded network".
- Queueing delay, which depends on the network load.

A segment loss slows down the connection as time outs and retransmissions take place. The throughput of a long TCP connection is approximately proportional to $1/\sqrt{p}$, where p denotes the packet loss probability [MSMO97]. Most HTTP connections, however, are short – in this measurement the median is around two kibibytes – i.e. a few TCP segments and the formula is not valid for those.

6.1.2 User Impatience in a Network

Web traffic (HTTP [BLFF96, FGM+99]) is carried on top of the TCP protocol [Pos81c]. In the normal case a TCP connection starts with sequence number synchronisation, where the initiating party (client) sends a TCP segment with

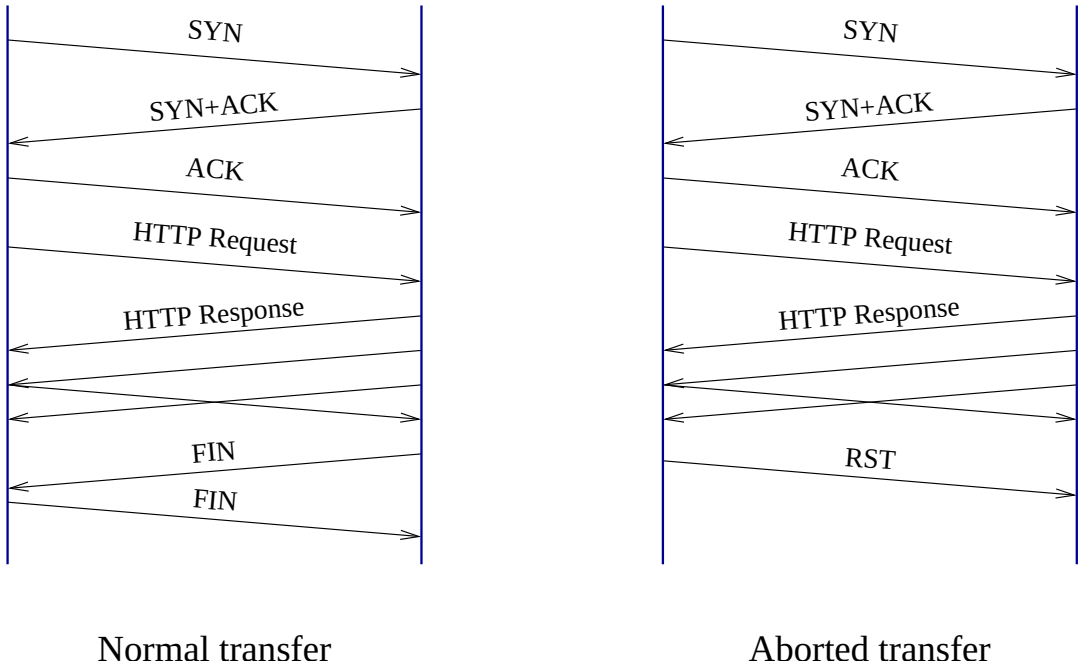


Figure 6.1: HTTP transfer in normal and aborted case.

the SYN flag set. The other party (server) replies to this with a segment having SYN and ACK flags set, which is then acknowledged by the initiating party. At this point the connection is set up.

When the TCP connection has been set up the HTTP request, including the headers and possible data, is sent to the server. The server processes the request and returns an appropriate response containing headers and document data. At this point the connection is closed if HTTP/1.0 is used. If both the client and the server support HTTP/1.1, more requests can be communicated on a single connection, or the connection can be left active to wait for more requests from the same server.

When the transfer is completed, the TCP connection is closed. In a normal case, the server sends a segment with the FIN flag set and the client acknowledges this and replies with a segment that has also the FIN flag set.

The user may get impatient, i.e. thinks that it will take a long time to complete loading the document, or the transfer has stalled. Most user interfaces have a mechanism to stop the transfer. Mainstream graphical browsers have a "Stop" button for aborting the transfer; the `Esc` key is bound to the same functionality.

When a connection is aborted, the client sends a TCP segment with the RST flag set. This shuts the connection down and the server will send no more data. If there is an intermediate caching proxy, this may continue to download the entire

document even if the user aborts the download and most of the document has already been received. In Figure 6.1 both a normal and an aborted connection are shown.

The connection is also aborted if the user selects a link from a partly loaded page. The browser aborts on-going connections and starts downloading the new pages.

By observing network traffic the following cases can be found:

(a) Items loaded completely. The TCP connection ends gracefully with the normal 3-way handshake.

(b) User aborts transfer. The TCP connection is aborted with a reset (the client sends a TCP segment with the RST the flag set).

(c) User selects a link from a partly loaded page. This one is like the case above, but a new connection is established soon after the aborted connection.

(d) Server aborts transfer. The server decides to abort the transfer for some reason.

For this study, the cases (b) and (c) are interesting: in both cases the connection is terminated with a reset. One can differentiate between these cases by looking whether new TCP connections originate from the same host shortly after the reset. In case (b) the aborted transfer is most probably wasted, while in case (c) not. For discussion identifying multiple transfers over HTTP/1.1 connection and other possible sources of error, see Section 6.2.1.

6.1.3 Abandonment Intensity

Define $F(t) = P[X \leq t]$ to be the CDF of the duration X of all the HTTP connections. Then let the $\lambda(t)$ be the ending intensity (hazard rate) for connections active at time t. These two are related by

$$\lambda(t) = \frac{F'(t)}{1 - F(t)}. \tag{6.2}$$

The following events are defined: S – the connection will end normally and R – the connection will be terminated with a reset. Using those events, the CDF for normally terminating connections can be defined

$$F_S(t) = P[X \leq t|S] \tag{6.3}$$

and $F_R(t)$ is defined similarly for aborted connections.

The ending intensity can be expressed as $\lambda(t) = \lambda_S(t) + \lambda_R(t)$, where the first factor is the ending intensity for gracefully ending connections and the second one for terminated connections. Since $F(t) = F_S(t)P[S] + F_R(t)P[R]$, it can be substituted into (6.2) and one deduces the ending rates,

$$\lambda_S(t) = \frac{F'_S(t)P[S]}{1 - F(t)}, \tag{6.4}$$

$$\lambda_R(t) = \frac{F'_R(t)P[R]}{1 - F(t)}. \tag{6.5}$$

6.2 Analysis Methodology

Both directions on a link are monitored and recorded on separate trace files. A part of traces used have a small but noticeable timing skew between the directions resulting from configuration error in the synchronisation between the capture cards. It would be possible to make an approximate matching up to a few milliseconds accuracy, but it was decided, however, to monitor only the acknowledgements. The focus is on what the client receives as the behaviour of the user is studied here, not of the network or server. This can best be seen from the acknowledgement numbers as the client acknowledges received data.

First all TCP segments that had port 80 as the destination port were selected. There exists also HTTP servers on other ports but they were excluded from the analysis for clarity. When a TCP segment with only the SYN flag set was seen, a new flow was started. Packet information (acknowledgement numbers, timestamps) was recorded for every flow where SYN was detected until a segment with either FIN or RST was found.

The HTTP/1.1 supports persistent connections, i.e. TCP connection is left open after an HTTP transfer to wait for more HTTP transfers from the same server. This would cause error in determining duration of the connection. For flows that ended normally (FIN), it was checked if any new data were transferred (acknowledged) within the last 10 seconds. If not, then the last segment that acknowledged

new data was selected as the end of the flow. It was also checked if the connection was a proper one and not missing a proper handshake.

The results were broken down by the client IP address and checked if there were new connections starting just after the reset. As the data had been checked, the CDF and the PDF were calculated. Based on those, the ending intensity was calculated.

6.2.1 Delayed Acknowledgement

One possible source of error would be delayed acknowledgements: a TCP receiver may not immediately send acknowledgement on receiving a TCP segment in order to limit rate of acknowledgement-only TCP segments. It may wait up to 0.5 seconds before sending an acknowledgement [BE89, APS99] but many TCP implementations do not wait more than 0.2 seconds [Pax97b].

The segments which have one of SYN, FIN, and RST flags set are not delayed as they are not acknowledgement-only segments. The only case when only-ACK segments are used in calculations is the case where closing of connection is delayed by more than 10 seconds after last data. As the delay of ACKs is well below a second, it does not have a great effect on transfer times over one second – the interesting ones.

6.2.2 Premature FIN to Close Connection

Some HTTP client software send a segment with the FIN flag set to abort connection instead of sending one with RST flag [Fel98]. Because of the timing skew between traffic directions, a premature FIN could not be reliably identified from a normal ending. This error results in *underestimating* the abandonment intensity but should not affect ending intensities on different time scales.

6.2.3 HTTP/1.1 Multiple Transfers

The use of HTTP/1.1 makes analysing document transfer times more difficult. A HTTP/1.1 connection may be left active after a document is transferred to wait for more transfers from the same site. This time may be short as in fetching images for the document or longer as a user reads the page and decides to click a link.

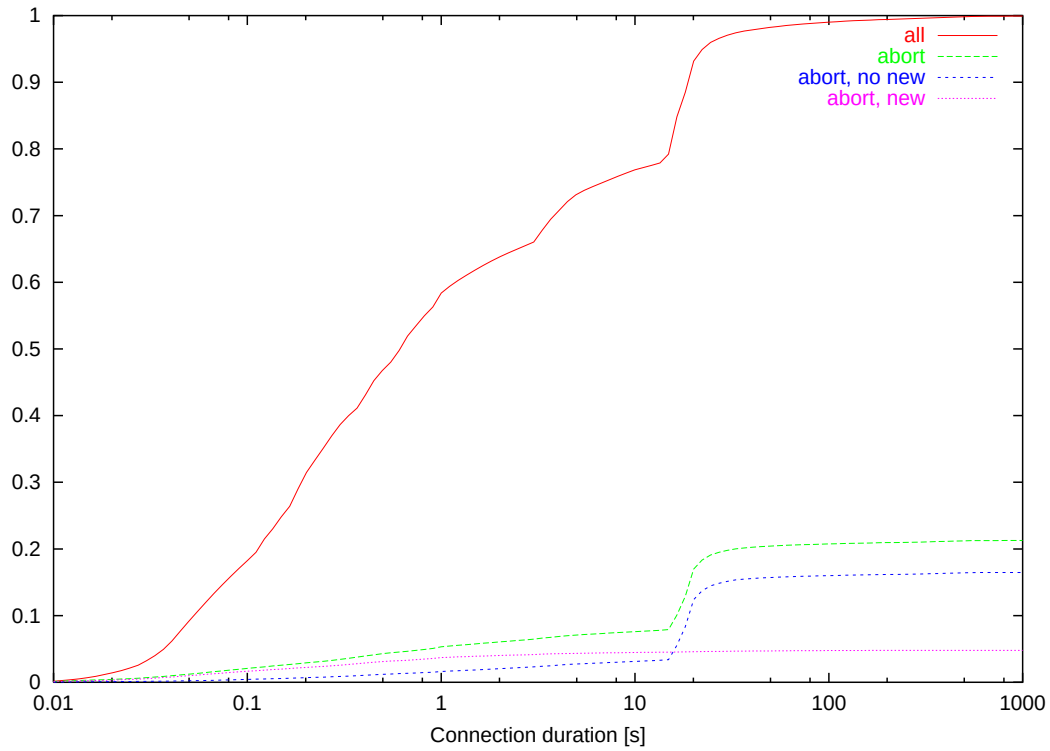


Figure 6.2: Ending intensity for dataset $\mathcal{I}$.

A multi-document transfer can be identified by looking at the sequence numbers and the acknowledgement numbers. In the beginning of a connection only the sequence number increases as the client sends a request to the server. When the server sends data to the client, only the acknowledgement number increases. If the client makes another request, then the sequence number increases again.

6.3 Results

Dataset presented in Table 3.1 includes only connections with one HTTP transaction identified as described in Section 6.2.3. These amount to about 80 % of all connections. Based on the measured data, the distribution of HTTP transaction count on a single TCP connection follows approximately the law bn^{-3} where b equals the number of connections with one HTTP transaction and n is the number of transactions. The distribution has a long tail.

In Figures 6.2 and 6.3 the ending intensities are shown as a function of the time for the datasets $\mathcal{I}$ and $\mathcal{E}$ respectively. The time scale is limited between 10 ms and 1000 s (16 minutes) as there are only few connections outside this range. In both figures there is a clear peak above 10 s for terminated (aborted) connections. This suggests that the “10 second rule” applies also on the web.

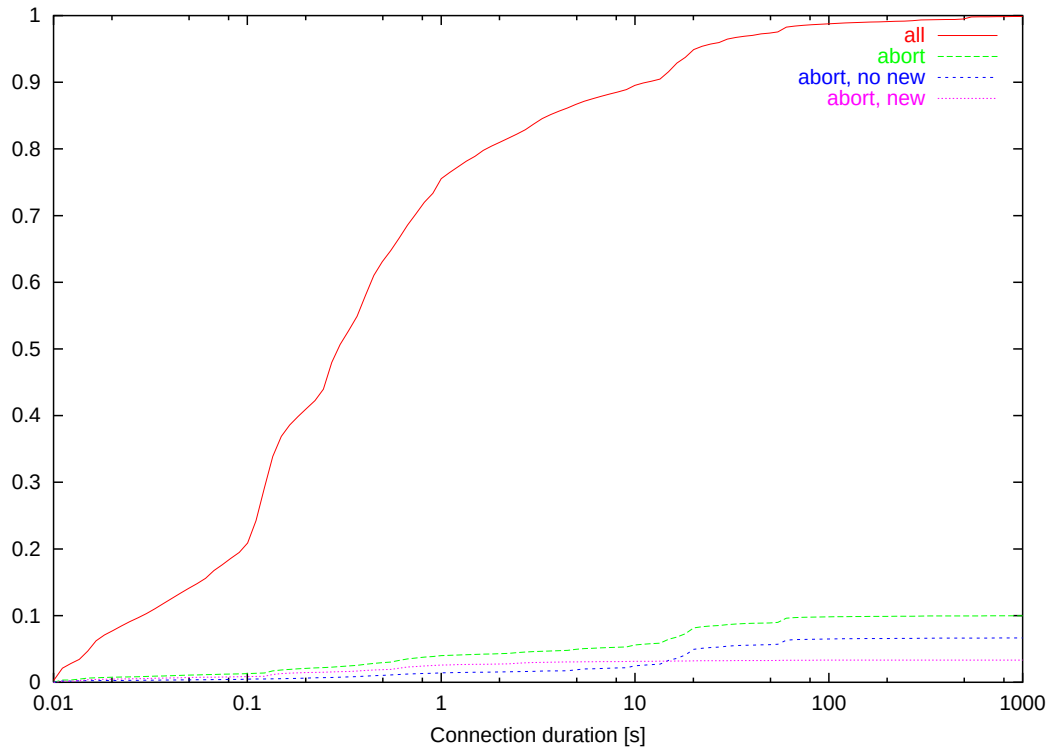


Figure 6.3: Ending intensity for dataset $\mathcal{E}$.

In dataset $\mathcal{I}$, 30 % of the aborted connections are followed by a new connection within 0.2 seconds. For the case $\mathcal{E}$, the corresponding figure is 20 %.

Based on the CDF over the connection lifetime for dataset $\mathcal{I}$ (Figure 6.4), it can be seen that at time scales below one second most of the terminated connections are followed by a new connection. This is due to the user selecting a link on a partly loaded page.

Connections lasting longer than one second were divided into two categories based on their throughput. The threshold was chosen as 8 kB/s (64 kbit/s) since a modem or ISDN user will get at maximum this bandwidth, depending on if compression is used or not. There is a clear indication that received bandwidth has an effect on the probability that a connection is aborted. In dataset $\mathcal{I}$, 29 % of the slower connections were aborted and 10 % of the faster connections. In dataset $\mathcal{E}$, approximately 10 % of the both connections were aborted. Corresponding ending intensity graphs are shown in Figure 6.5 and Figure 6.6.

The abandonment intensity of slower connections increases rapidly around 10 seconds, while faster connections are not aborted at the same rate. At one second, 8 % ($\mathcal{I}$) and 9 % ($\mathcal{E}$) of the faster connections are already aborted while only 2 % of the slower ones. It can be concluded that aborts of faster connections result from selecting a link and slower connections are aborted because of impatience.

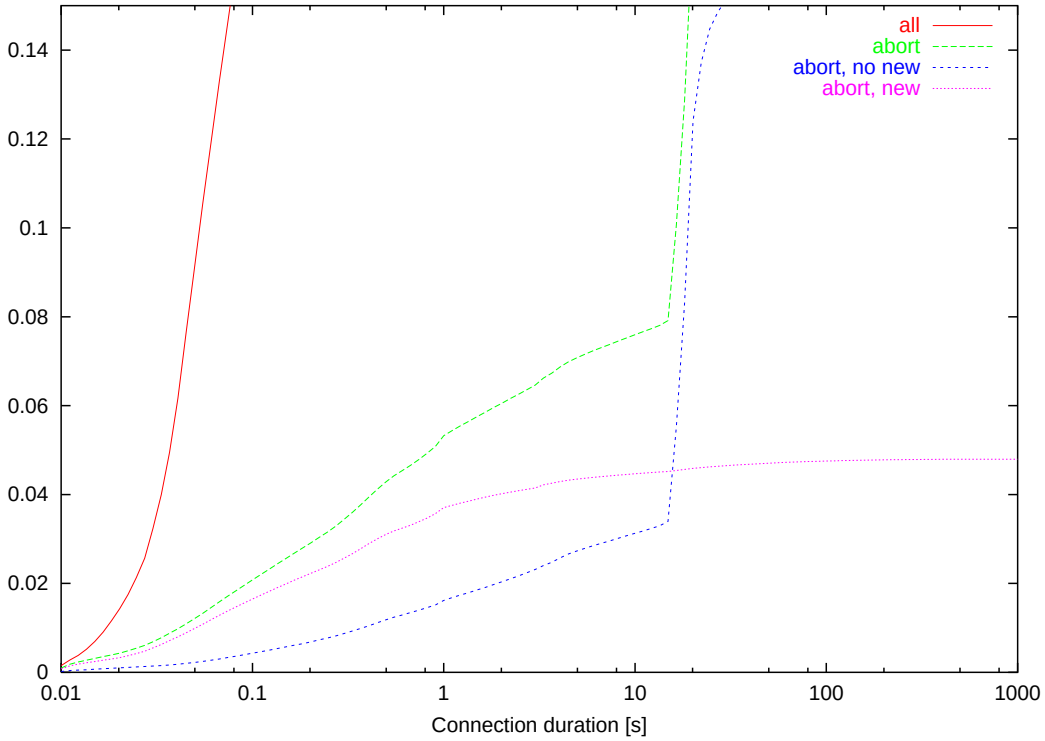


Figure 6.4: Connection duration CDF for dataset $\mathcal{I}$.

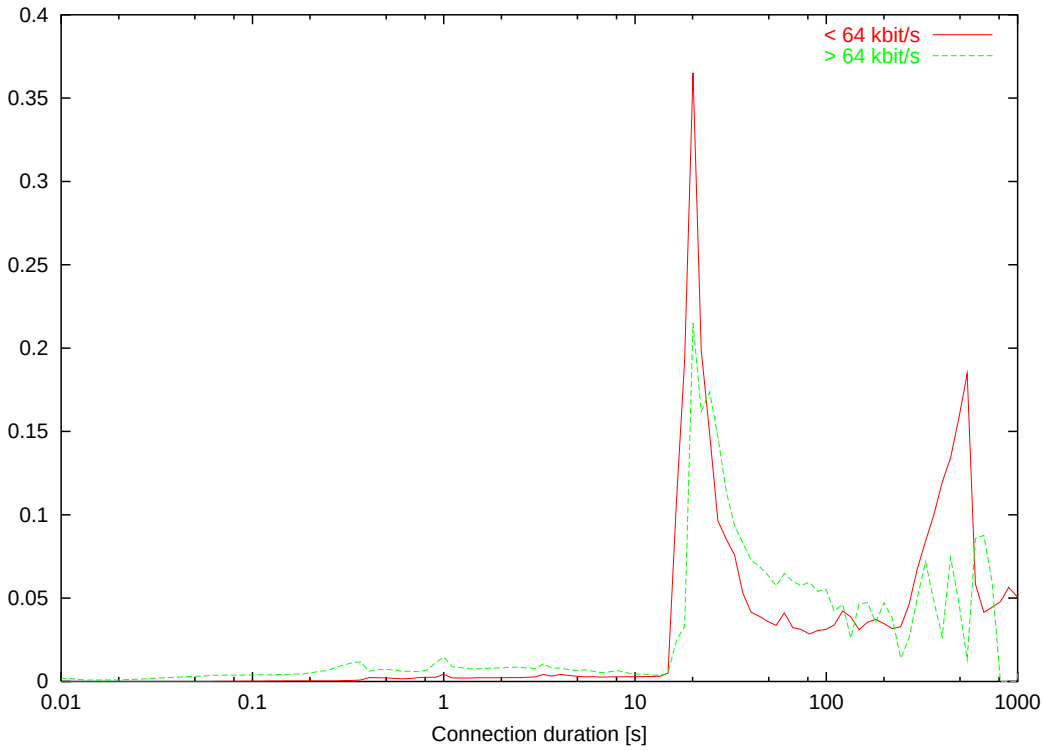


Figure 6.5: Ending intensity for dataset $\mathcal{I}$ for different throughput.

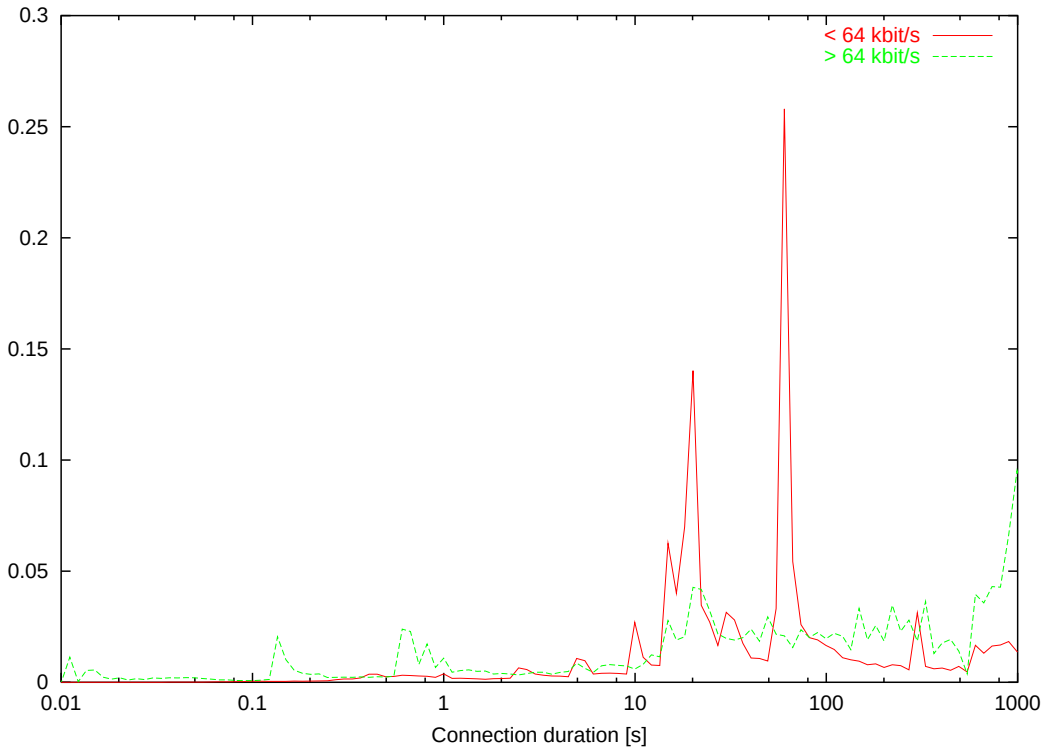


Figure 6.6: Ending intensity for dataset $\mathcal{E}$ for different throughput.

The transferred file size did not have as clear effect as the throughput. Files were divided into two categories: web page components, which are less than 64 KiB and downloadable files which are larger. There was no clear trend except the anticipated one: the maximum of the abandonment was at longer time scales with larger files (see Figure 6.7).

It was also studied if the time of day has any effect on the user impatience. The day was divided into three periods: morning (7 am to noon), afternoon (noon to 5 pm), and night (5 pm to 7 am). No significant differences in the abandonment intensity were observed in either dataset.

6.4 Conclusions

Based on measurements the abandonment intensity for the web (HTTP) traffic was studied. A user was considered impatient if the TCP connection was terminated with a reset (RST). A sharp increase in the abandonment intensity occurs if the connection lifetime is more than 10 seconds. From the data it was also shown that the connection bandwidth does have an effect on abandonment intensity.

Based on the results, it can be concluded that to avoid a user becoming impatient

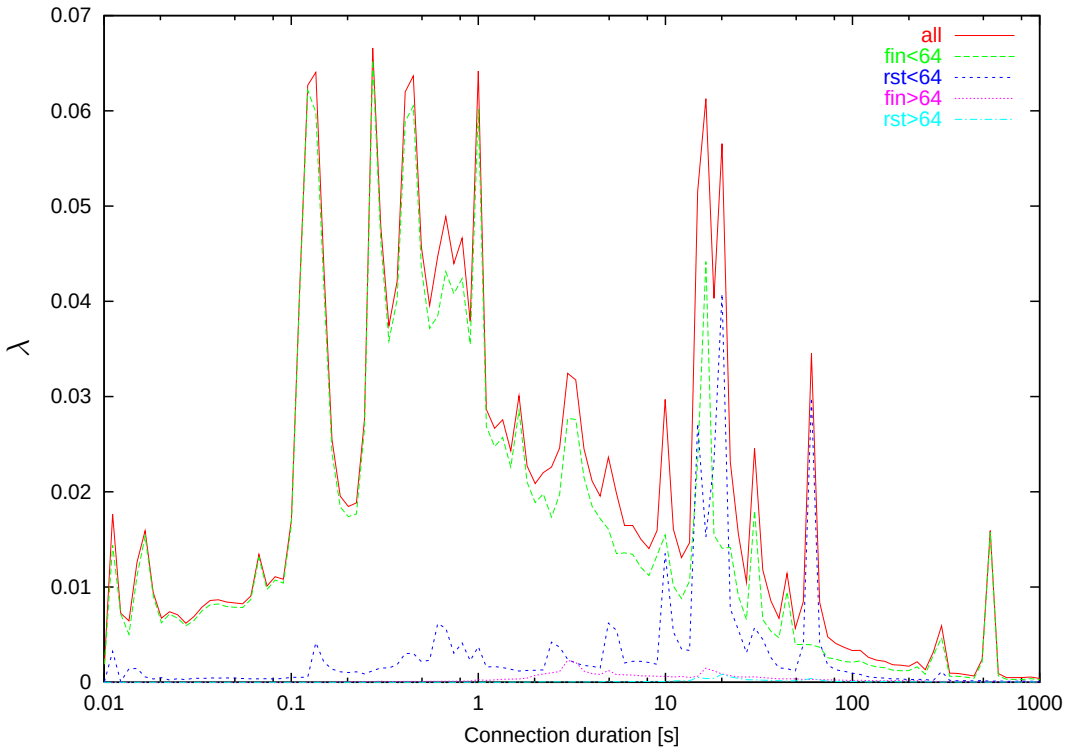


Figure 6.7: Ending intensity for dataset $\mathcal{I}$ for transfer size.

it is important to have web pages loading in less than ten seconds. Also the throughput must be satisfying: slow connections had a greater probability to get aborted than the fast ones.

In the measurement location it was possible to study traffic both near a client and near a server. Different profiles were noticed as the server (dataset $\mathcal{E}$) was mainly accessed from the local campus network (10 % of connections) and other locations with fast connections. The local clients ($\mathcal{I}$) accessed servers all over the Internet and received slower connection speeds on average.

User impatience measurements can be utilised by a network operator to monitor the actual quality of service the users are experiencing. Compared to simple delay, throughput, and packet loss measurements, the impatience measurement automatically adapts to user's behaviour and has a clearer relation to the actual quality of service.

Studies on the user behaviour would be useful in the future, in addition to network monitoring and measurements. One possibility is to a use an instrumented computer to record user actions and software responses and to correlate these with network measurements. More measurements from the network are also needed.

Chapter 7

Estimating available bandwidth on network

Traffic volume in the Internet varies much on all time scales. This has been demonstrated by multiple measurements and use of measurement-based call admission control (CAC) could easily result in network overload or waste of resources [BJS00]. There are, however, proposals to use *probe*s to evaluate if there is sufficient capacity in the network.

Another application of bandwidth information is server selection. If there are multiple servers carrying the same content, a client – and possibly the network – would benefit from selecting "nearest" server. There are several proposed methods to select server. In [CC96a] probes are used to estimate bottleneck link speed, available bandwidth and latency. For each server a predicted transfer time is calculated by taking into account document size and measured figures.

Both for estimating sufficient network capacity and locating optimal server, it is critical that the network is stable enough. The lifespan of stable time is dependent on the application. A telephone call would need stability for a few minutes on the average, while Internet radio needs stability possibly for several hours.

7.1 Study of network throughput stability

First the network traces were analysed as described in Chapter 4. From all network flows with a timeout no longer than 60 seconds, bulk transfers with more than 65,535 bytes, including all header data, were selected. A bulk transfer was defined

to be one which had average packet size more 512 bytes and had variance less than two times of average packet size.[1]

For each flow the *bandwidth* received was calculated, see Section 1.2 for the definition of bandwidth. All flows that matched to the criteria were grouped by host pairs, i.e. transfers from one host to another formed one set. For each set, flows were sorted by their start time and the difference between the end of a flow and the start of the following flow was calculated. One should note that this time may be negative if the flows have been active at the same time. In this case the time difference was set to zero.

Intra-campus host pairs accounted for approximately 18 % ($\mathcal{I}$) and 11 % ($\mathcal{E}$) of all host pairs. Those host pairs had 13 % and 6 % of flow pairs in dataset $\mathcal{I}$ and $\mathcal{E}$ respectively. A typical intra-campus connection is between two hosts that are connected via fast LANs and are only few routers away from each other. One can presume that the end system load has a major effect on bandwidth in comparison with network effects in this case. For the rest of this network stability study, all intra-campus flows were ignored.

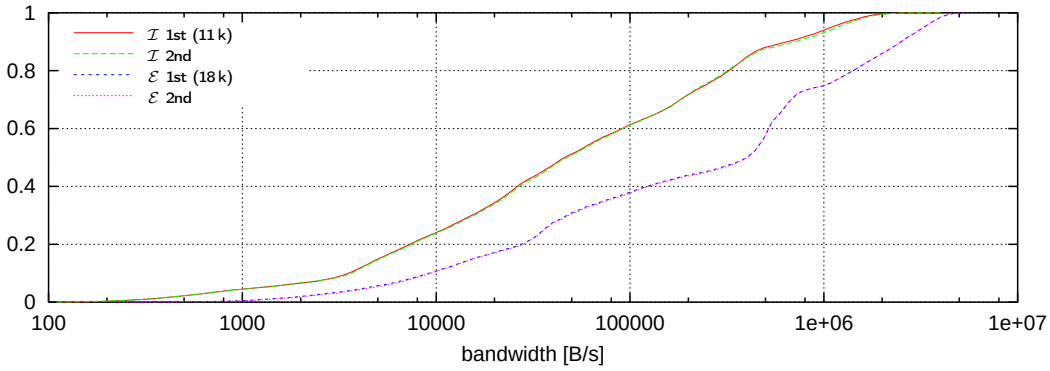


Figure 7.1: Cumulative distribution on received bandwidth.

For dataset $\mathcal{I}$ nearly all selected flows (98 %) were HTTP transfers while for dataset $\mathcal{E}$ HTTP contributed 85 % of all flows. Other applications include FTP, HTTPS and IMAPS.

Figures 7.3 and 7.4 show the cumulative distribution of the ratio between two consecutive flows for time intervals less than ten seconds, less than one minute and less than five minutes. Shorter intervals were excluded from the longer interval sets.

Approximately 60 to 80 % of the flow pairs in dataset $\mathcal{I}$ with less than five-minute

[1]This criteria assumes that there are two 40-byte packets at start of flow and two 40-byte packets at end of flow. If other packets are approximately same size as average (i.e. maximum size used by TCP implementation in question), then variance approaches this limit.

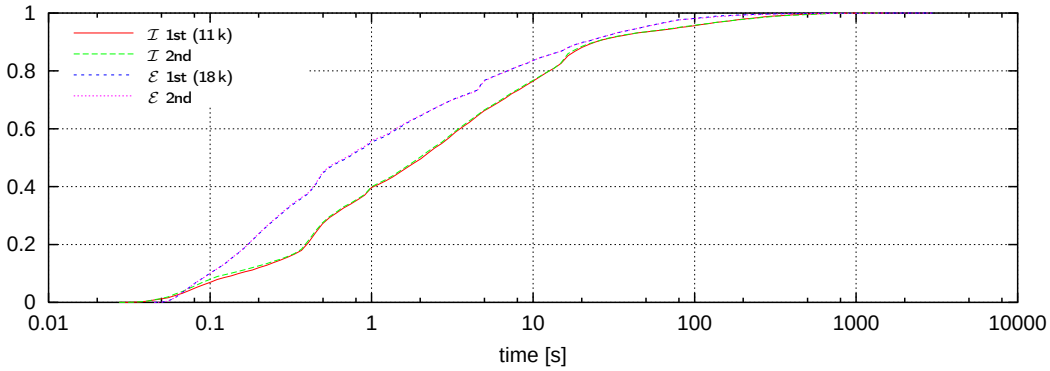


Figure 7.2: Cumulative distribution of transfer time.

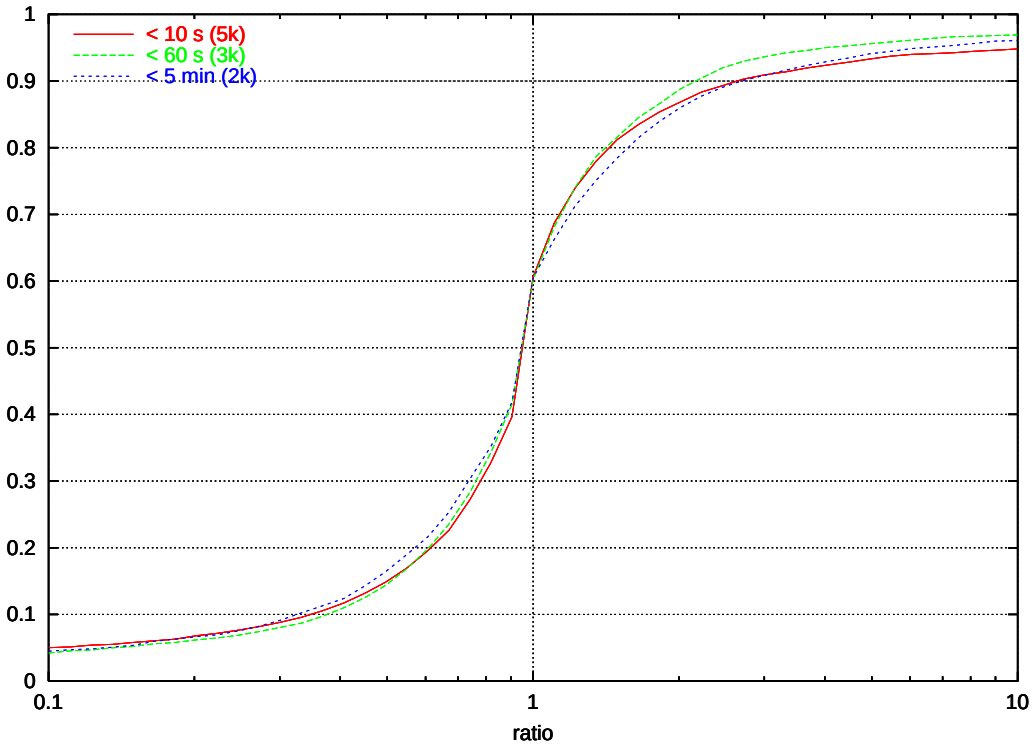


Figure 7.3: Flow pair bandwidth ratio for dataset $\mathcal{I}$.

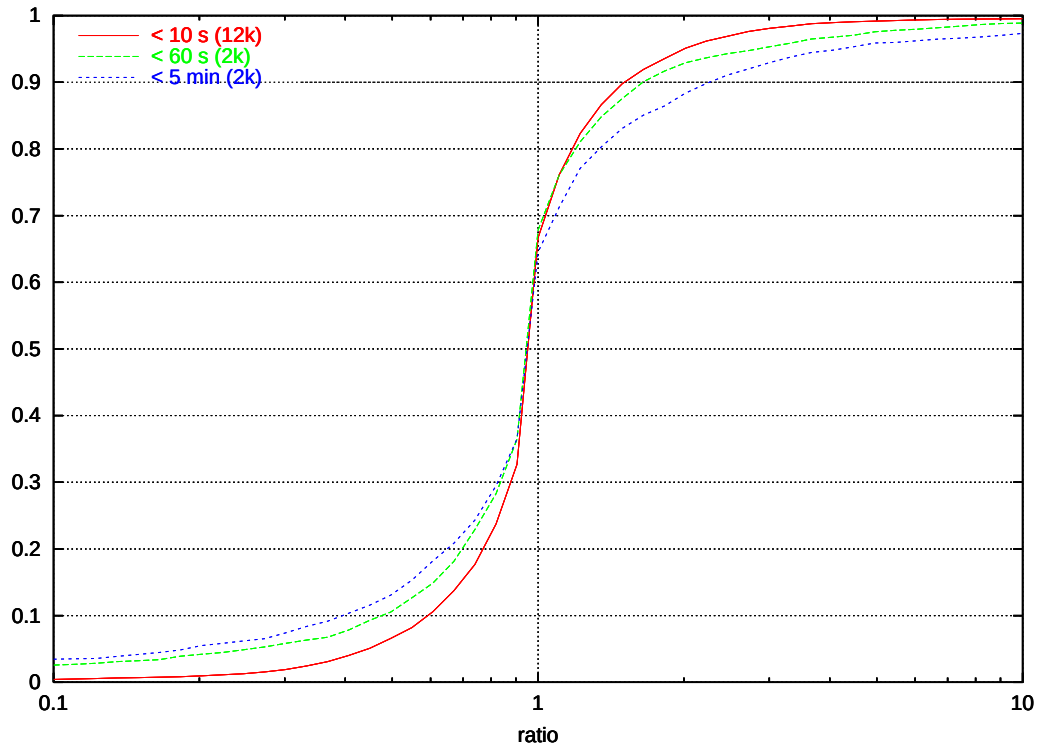


Figure 7.4: Flow pair bandwidth ratio for dataset $\mathcal{E}$.

interval did receive similar bandwidth, i.e. the bandwidth received by the other was not more than twice that received by the other one. In dataset $\mathcal{E}$ shorter intervals performed better but performance was degraded for intervals over one minute.

If the flow interval is allowed to extend full range of measurement period with ranges of one minute, five minutes and over five minutes, it can be seen from Figure 7.5 that subsequent flows are the more dissimilar the longer the time interval is.

7.1.1 Bandwidth correlation in function of interval

It was studied if the correlation between bandwidths of the first and the second flow are related to interval. Same classification (10 s, 1 and 5 minutes) was used. There was not any significant trend found using linear correlation. When the linear correlation was applied on logarithm of bandwidth, the correlation coefficient for $\mathcal{E}$ reduces from 0.94 via 0.84 to 0.78 when interval was increased. However, for dataset $\mathcal{I}$ the trend was opposite from 0.57 to 0.79.

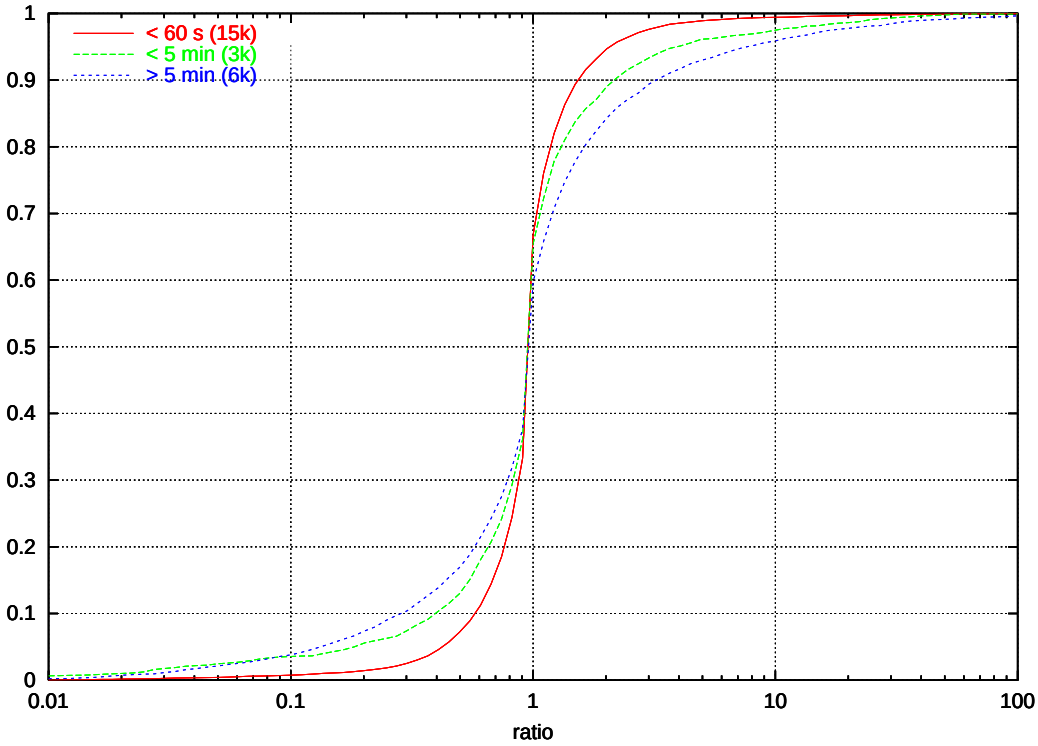


Figure 7.5: Flow pair bandwidth ratio for dataset $\mathcal{E}$ over extended period.

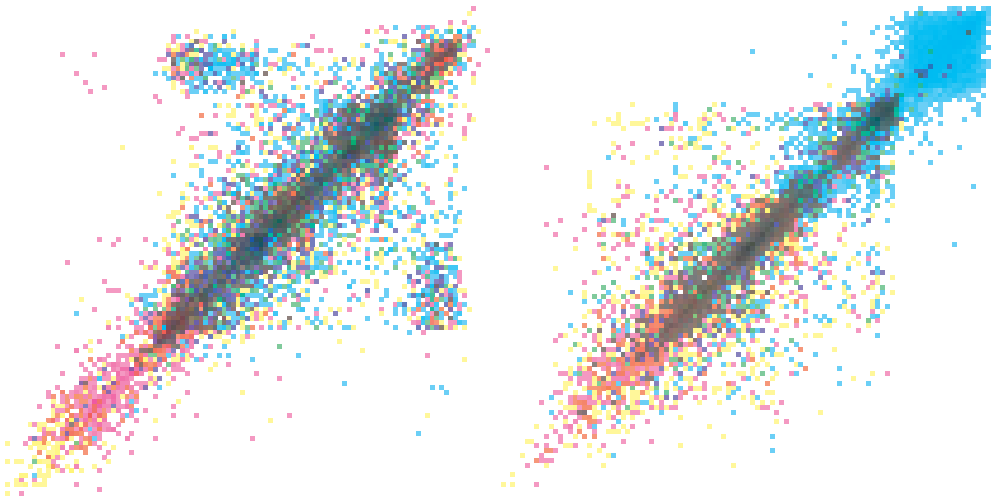


Figure 7.6: Bandwidth of probe pair in phase density plot. Probe intervals are represented by colours: cyan is for flows less than 10 seconds intervals, magenta for 10–60 s and yellow for 1–5 minutes. The darker the colour component, the more there are occurrences. Dataset $\mathcal{I}$ on the left (110 B/s – 4 MB/s) and $\mathcal{E}$ on the right (220 B/s – 5.4 MB/s), logarithmic scale.

7.1.2 Conditional selection of probe pairs

If one considers a typical application for bandwidth probe, real-time communication, the needed goodput does have some upper limit, a threshold. If there is more bandwidth available, it does not anymore benefit the application.

Bandwidth stability requirement can thus be expressed in another form: *"if a flow receives bandwidth X bit/s (or more), what is the probability that the following flow will have bandwidth of X bit/s or more"*. The X depends on application. Typical values for would be 22.5, 64 and 384 kbit/s, which would correspond to GSM-grade voice, good quality voice and high quality audio or low quality video, respectively.

According to results in Table 7.1, a success rate more than 94 % can be achieved for low-bandwidth flows. When bandwidth demand is increased, the success rate becomes lower and this trend continues for higher bandwidth requirements. For example, in dataset $\mathcal{I}$ the success ratio is less than 66 %, if the required bandwidth is more than 1.5 Mbit/s (approximately 9000 observations, 15 %).

Table 7.1: Conditional bandwidth probe success rate for TCP probes.

Target bandwidth	22.5 kbit/s		64 kbit/s		384 kbit/s	
Dataset	$\mathcal{I}$	$\mathcal{E}$	$\mathcal{I}$	$\mathcal{E}$	$\mathcal{I}$	$\mathcal{E}$
Number of pairs	10,333	17,368	9,190	16,651	5,900	13,059
Proportion of pairs	96.4 %	98.2 %	85.8 %	94.1 %	55.1 %	73.8 %
Success rate < 10 s	99.6 %	99.8 %	95.0 %	99.1 %	84.1 %	98.2 %
Success rate 10 – 60 s	99.1 %	97.9 %	94.4 %	94.4 %	85.9 %	87.7 %
Success rate 60 – 300 s	98.7 %	96.7 %	93.1 %	92.1 %	82.8 %	80.6 %

Investigating the success rate is not sufficient to determine the value of a method. One must know also the probability of false negatives because a high success rate may come with a high ratio of false negatives. A false negative is a case where the first connection fails to satisfy bandwidth requirement but the second connection receives sufficient bandwidth. Proportion of false negatives was high in low-bandwidth cases, reaching up to 72 %. For higher bandwidths, the probability of false negatives was in the same range as that of false positives.

7.1.3 Possible problems in methodology

The flows used to estimate available bandwidth were real application TCP flows. The majority of those were HTTP connections but they included also FTP and other protocols. This approach has several issues that should be carefully studied.

First of all, host pairs and times when transfers took place were not artificially selected *a priori*. Instead, they resulted from normal daily network usage and thus are a representative sample of destinations and times one would expect for an application using probes.

Secondly, they are non-realtime application flows. Traffic characteristics are different from a real-time application. There may be some application-dependent delays that are not caused by the network. These include the time used for the initial TCP handshake. However, if delays are caused by end system performance problems, they may cause problems for a real-time application also.

Besides, the bandwidth is measured over the whole flow lifetime. There may be significant fluctuations over time period that may result in severe degradation for real-time applications.

Lastly, the probe selection used considers only probes by their start time order. If there are multiple flows between two hosts within time period, a flow is compared only to the nearest one by start time.

From the above we can notice that there are issues, which work to opposite directions considering results. Based on this material it is difficult to determine which factors are the most significant ones. These measurements could be accompanied by active measurement as in [Asa98].

7.2 Conclusions

Based on the measured data, flow throughput was not found stable. If the interval between probes was more than one minute, one probe had a bandwidth more than twice as large as the other one in the majority of observations. It can be concluded that predictability of the maximum bandwidth is low for any significant interval between flows. Utilising probes for file transfers may not be worth extra effort if connections do not have very dissimilar properties.

Using threshold values for admitting probes gives better results and outperforms random selection. However, its failure rate at higher data rates and longer interflow times is more than 20 %. This ratio cannot be considered as adequate to give guarantees for end user media fidelity. Furthermore, there are a significant number of false negatives for low bandwidth probes.

Chapter 8

Need for network measurements

At the first sight, measuring network characteristics by observing network traffic looks like doing things the hard way. After all, a network is man-made system whose component characteristics are known so it should be possible to construct an exact model for simulations. However, there are several reasons why this is not as trivial as it looks.

Network operators have knowledge of their network structure and configuration, but this information is usually for their own use only and not published. Network topology, physical locations and detailed information, such as equipment hardware and software versions and configuration parameters, are kept secret both for security and competitive reasons. Furthermore, several changes take place in networks daily, making it even more difficult to acquire consistent snapshot of the network. The network characteristics are also changed abruptly by failures, which are difficult to model because of their random nature and unknown parameters.

Another unknown factor in networks is the characteristics of network traffic. New applications emerge and proportion of each application of total traffic changes. Network users adopt quickly new applications if they find them useful. Even distribution of traffic flow directions may change, the latest example being peer-to-peer sharing applications where communication is mesh-like rather than server-centric. Additional traffic sources are malicious worms, which spread either automatically or human assisted utilising security problems in software [SPW].

8.1 Operator's view to measurements

A network operator can use measurements for two purposes: network monitoring and accounting. The first one is important to operator to maintain network performance. Network performance monitoring can be further divided into three different areas: network failure identification, traffic load monitoring, and information gathering for upgrade needs. While operator should receive timely alerts from network management system in case of link or equipment failures, there is a group of network problems, which are not related to these. One example of such a problem is a routing problem, where a network routing database is inconsistent or otherwise erroneous. Traffic load monitoring is important in order to promptly react on increased network traffic, which may be a sign of other problems.

Operators have also different roles. Some operators, "tier-1", operate transit backbone networks that span over a country or are global. These operators have peering agreements with other tier-1 operators and multiple tier-2 operators. Tier-2 operators are regional network service providers that serve big corporate customers and small Internet service providers. Tier-2 operators may have private peering with other tier-2 operators and may connect to two or more tier-1 operators.

Tier-3 operators provide Internet services for end users and organisations. One has tens of thousands of customers and thousands network nodes to operate. A tier-2 operator has fewer customers and network nodes, and a tier-1 operator may have only hundred or so nodes [Wii01]. This will have an impact on how network should be monitored. If one has tens of thousands of low-bandwidth links, deploying special hardware on each link is much more difficult than if one has only tens of links, even if they are high-bandwidth ones.

Further, tier-3 operators may have different customer bases. Some of operators have more focus on corporate customers, some provide services for private users. It is possible to provide only access services and maybe some infrastructure such as name service and email while some provide full service including service hosting platforms [Hus98]. Those factors have an effect on how custom services on can provide: if there are hundreds of thousands private customers, one cannot provide as individual service as for some tens of large corporate customers.

8.1.1 Network dimensioning by measurement data

One of the key factors for the success of an operator is a proper dimensioning of the network. An under-provisioned network results in unsatisfied users and lost revenues while an over-provisioned network may be too expensive to deploy or has too high leasing costs. An operator should find the optimal time for upgrade considering the present capacity and the growth of traffic demand and, on the other hand, development of hardware and its costs, including its deployment. Cost of the physical transmission medium, such as "dark fibre", including its installation depends only marginally on transmission speed it is used for.

8.1.2 Usage accounting

There are a few reasons an operator would want to monitor traffic by individual customers. First of all, the operator wants to monitor network for possible abuse. If there is a sudden increase in traffic originated by one customer, there may be some problems at that customer's system, for example a break-in. By monitoring each customer, it may be possible to notice anomalies before they disturb the network at large [Hus98].

Secondly, there may be monthly traffic limits in agreement between a user and an operator. To monitor that the user does not to exceed those limits, the user traffic must be measured. This kind of measurement is easy to accomplish, especially if the user is connected with a fixed line.

An operator may have different costs for different paths. Traffic that is internal to an operator does not cause any excess costs for the operator if link capacities are sufficient. This traffic could easily be charged with a flat rate. Other targets may then have different costs. For example, the operator may have to pay for an upstream network provider for domestic and global traffic by monthly volume.

In order to link its own expenses to user fees, an operator needs to monitor traffic: which portion of traffic by a customer is local, domestic, global, or served by cache systems.

8.1.3 Security related monitoring

Considering the inherent insecurity in IP networks, continuous need to monitor network traffic is acknowledged also by operators, not only by network clients.

Certain attacks such as DDoS are noticed by network traffic increase but there are many attack types, including some of DDoS attacks, which do not result in excessive traffic into the network but render the end systems unusable. These attacks are identified either on end systems or by the IDS monitoring network.

In the IP header there is no trusted information: each field is filled by the sending end system and possibly modified by intermediate routers and gateways. The true origin of an IP packet cannot be found from the header fields. This makes it difficult to locate the source of intruding packets as strict ingress filtering [FS00] is not deployed universally. Without advanced tracing support by routers it is practically impossible to find out where packets came from; especially if the attack is short in duration. There are no connection records in a packet-switched network.

There are several proposals to facilitate traceback on the Internet [SWKA00, SPS+01]. Each of them requires modifications to the router hardware in order to work without degraded forwarding capacity. It is most probable that in the first phase these are deployed as external devices at critical links.

8.2 User's view to measurements

Users do not initially have much reason to measure network traffic. They are not interested in actual performance figures, such as per packet delay or packet loss, but in how applications perform: do they, or others that use their services, have an acceptable fidelity.

In the traditional telephone network all the components, including end systems such as telephones and PBXs, were managed by a small number of operators. The majority of calls were originated and terminated inside a single operator's network. If there was some issue related to the grade of service, it could be resolved by a single operator. The operator could not blame anybody else.

The current Internet is a very different environment. There are multiple operators and most of the connections traverse across multiple networks. Current statistics (2001 December) indicate that there are more than 12,000 autonomous systems (AS) in the Internet announcing more than 100,000 routing prefixes [BNkc02]. AS path lengths, i.e. number of ASs needed to transit packets to the destination network, were mostly around 3 to 5 from selected backbone networks with a maximum of 11.

8.2.1 Service level agreements

A customer subscribes to a service provided by a network operator and the customer expects to receive the service as planned. It may happen, however, that the operator fails to deliver service to the customer satisfaction. In the telephone network, failures are principally easy to identify: the user either receives a dial tone or does not and dialling is successful or not. Problems in voice quality are rare due to digital transmission but they may occur in cellular networks.

For packet networks there are three fundamental factors that define service fidelity: available bandwidth, packet loss rate, and packet delay. These contribute to the service level, which can be satisfactory, degraded, or unavailable. Threshold values for above factors for each service level depend on application used. A service level that is quite sufficient to send and receive emails may not be satisfactory for browsing multimedia content or for VoIP calls.

Some operators define only two levels of service: available and unavailable. The service availability verification "is accomplished by the Operator pinging the Customer's router" [Ear02]. If the user is paying premium rate for a better service, one undoubtedly wants to evaluate if the investment made is cost efficient [FH98].

If the service level agreement is viewed on user's side, the user is not interested in network metrics but application metrics; for example application response time is the most important factor for online systems [Stu01]. If the operator measures only "network metrics" one may fail to identify some application-dependent problems. From the operator viewpoint the network is performing according to objectives stated in the agreement but the user may still be unsatisfied.

8.2.2 Application measurements

Applications can be instrumented for measuring the performance user receives. The RTP protocol provides performance data exchange using its companion protocol RTCP. RTCP includes sender and receiver report records, which have information on the network performance. These performance figures include the number of lost packets, the delay variation and the last packet received. Each host transmits periodically these reports to the same multicast group as the original transmission and all members of the group thus receive those. Each sender and receiver can then compare its own performance to that of the others and the network operator can do it as well.

The RealPlayer by RealNetworks, which uses a proprietary protocol to transmit real-time audio, video and other media, includes also a mechanism to give feedback and statistics to the originating server. This information exchange can also be denied by the user settings.

It is also possible to instrument end systems. In [Peu97] a Linux kernel was instrumented to record essential information about network protocols and buffer occupancies at operating system level. It is also possible to monitor user actions and measure times between each action, for example to determine the time needed to connect to a database [Com].

End system instrumentation is, however, labour intensive if the monitoring is not included into application software. Properly organised, client-side measurements can provide important details, which are difficult to obtain by monitoring network traffic.

8.2.3 Information provider measurement objectives

From the network performance point of view, an information provider wants to deliver data to users by optimising network capacity and costs of the network. The main question is "are my users satisfied with the quality they receive". If there are some problems, the information provider wants to find out if one can do something to remove those.

There may be problems in service, such as too complicated page layouts, problems in navigation or discrepancy between provided media and typical end system capacity. Differentiating between problems caused by the service and problems caused by too slow or otherwise inferior network connections can be difficult. In some cases these problems are related as the users' network capacity is not sufficient to transfer media content timely.

8.2.4 Organisational user

An organisation, which uses Internet for its internal communication or communication with its partners does usually have strict requirements for network availability and performance. The situation is different from an information provider as the organisation usually has a finite number of important sites that must receive sufficient performance.

It is possible to set up test systems at different sites that access service on other sites and measure performance. If performance falls below a set threshold value, an alarm will be posted. It is also possible to instrument user workstations to automatically measure the performance. For example, a background task could measure how long it takes to connect to a database server. Again, an alarm could be generated if the time is too long.

8.2.5 End user measurement objectives

An end user does not usually actively measure the network performance. For many users, the only performance figure they see is the speed of a file download displayed in the status window of a web browser. Other indicators one may notice are status displays of streaming media players that may report packet delay and packet loss rate with green, yellow, or red indicator.

If the network is used for non-critical tasks and the network has degraded performance, the user may stop using the network and come back later. If this happens regularly, dissatisfaction with the operator will increase.

8.3 Conclusions

Network measurements do have their role, both for the operator and for the user. The measurement objectives are different: the operator views his network as a set of routers and links whose performance the operator wants to optimise. The operator may measure link utilisation, error rates, loss rates and delays. These provide a view to the network and these will identify a set of problems.

The view to the network is very different by the user who does not see individual components but the end-to-end performance of his applications. Even if the average performance of the network is satisfactory, for example the loss rate is below some limit, an application's performance may be inferior. It is also possible that the problems are not in the network but at the end systems. It may be difficult to make a difference between these two cases.

There is a need for performance metrics that would reflect performance experienced by the users without adding an extensive amounts of test traffic to the network. The user impatience studied in Chapter 6 may be one of the measures an operator could utilise.

Chapter 9

Conclusions

In this work network measurements were carried out over a period of 18 months, see Section 3.3 for the key figures. In order to solve problems in long-term trace management, an efficient compression method for measurement trace archive was developed with a possibility to remove sensitive information from network traces.

Analysis methods were developed to study measured traces. First, network applications were classified into a few classes based on their specifications and experiences based on implementations and application usage. Some representive traces were annotated as examples to show characteristics of application protocols.

Secondly, network traffic was classified into flows. Flow properties of different applications were studied. The effect of different flow definitions was studied, especially that of flow granularity and timeout. Previous findings about different timeout sensitivities were confirmed [CPB93]: some of the applications are very sensitive to the length of the time-out period. Also there was a difference between applications depending if the complete 5-tuple was used or if destination port was ignored. In some protocols, such as HTTPS, there could be even a two orders of magnitude difference.

Selecting proper granularity and timeout values for each application can reduce the amount of information to be collected and processed. A long timeout, on the other hand, increases the state information a network element must maintain. This may be a problem in the core network.

User impatience is one of factors affecting the proportion of useless traffic in the network. If a user abandons loading of document before it is completed, the transferred data has used network resources gratuitously. The time needed to transfer a document was found to have a significant effect on abandonment

intensity: if downloading took more than 10 seconds, the transfer had an increased risk to be aborted. A low goodput was also found to be a reason to abort a connection. The document size cannot be attributed as a factor for abandonment by its own. The user behaviour appeared to be independent of the time of day.

The utility of real-time applications such as VoIP or streaming media in part depends on the network throughput stability. Based on measured flow throughput, this was studied. It was found that for any significant period of time (a few minutes), the network stability is not good enough for a probe to guarantee satisfactory throughput.

Finally, the need for network measurements was discussed. It was concluded that an operator and a user have somewhat different objectives for measurements. It may happen that the operator and the user disagree about the available service level and still both can be right. The operator may measure valid metrics from network operations monitoring but those metrics could fail to measure the network characteristics relevant to the application the user is using.

It is important that proper measurement methods are developed and deployed that enable an operator to monitor the network performance as a user of an application experiences it. Only then right measurements are being carried out.

Bibliography

[AC89] P.D. Amer and L.N. Cassel. Management of sampled real-time network measurements. In *Proceedings 14th Conference on Local Computer Networks, 1989*, pages 62–68, October 1989.

[ACTW96] J. Apisdort, K. Claffy, K. Thompson, and R. Wilder. OC3MON: Flexible, affordable, high performance statistics collection. *Proceedings of the Tenth Systems Administration Conference (LISA X) (USENIX Association: Berkeley, CA)*, page 97, 1996.

[ADPCH+92] F. Alvarez Del Pino, R. Chow, S.F. Hussaini, H.A. Latchman, and G.K. Madhusudan. Performance analysis and traffic characterization of an ethernet campus network to identify and develop possible smds applications and scenarios. In *IEEE Southeastcon '92, Proceedings*, volume 1, pages 398–391, April 1992.

[AKZ99a] G. Almes, S. Kalidindi, and M. Zekauskas. A One-way Delay Metric for IPPM. Request for Comments RFC 2679, Internet Engineering Task Force, September 1999.

[AKZ99b] G. Almes, S. Kalidindi, and M. Zekauskas. A One-way Packet Loss Metric for IPPM. Request for Comments RFC 2680, Internet Engineering Task Force, September 1999.

[AKZ99c] G. Almes, S. Kalidindi, and M. Zekauskas. A Round-trip Delay Metric for IPPM. Request for Comments RFC 2681, Internet Engineering Task Force, September 1999.

[Ano95] Anonymous. *FIPS 180-1, Secure Hash Standard*. National Institute of Standards and Technology, US Department of Commerce, Washington, DC, USA, April 1995.

[ANS96] ANSI/IEEE. Information technology–telecommunications and information exchange between systems–local and metropolitan area networks–LAN/MAN-type specific requirements, part 3: Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications and 802.3u-1995 supplement to IEEE std 8802-3 : 1996: Media access control (MAC) parameters, physical layer medium attachment units, and repeater for 100 mb/s operation, type 100baset (clause 21-30). ISO/IEC standard, ISO/IEC, 1996.

[AP99] Mark Allman and Vern Paxson. On estimating end-to-end network path properties. In *SIGCOMM'99*, September 1999.

[APS99] M. Allman, V. Paxson, and W. Stevens. TCP Congestion Control. Request for Comments RFC 2581, Internet Engineering Task Force, April 1999.

[Asa98] M. Asawa. Measuring and analyzing service levels: a scalable passive approach. In *1998 Sixth International Workshop on Quality of Service, 1998. (IWQoS 98)*, pages 3–12, May 1998.

[Bar68] P. Bardy. A statistical analysis of on-off patterns in 16 conversations. *The Bell System Technical Journal*, 47:73–91, January 1968.

[Bar94] Elmar Bartel. New TTCP program. On web, `http://www.leo.org/~elmar/nttcp/`, 1994.

[BBC+98] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Service. Request for Comments RFC 2475, Internet Engineering Task Force, December 1998. (Informational).

[BC95] Hans-Werner Braun and K. Claffy. post-NSFNET statistics collection. In *Proceedings of INET'95*, pages 577–587, June 1995.

[BCS94] R. Braden, D. Clark, and S. Shenker. Integrated Services in the Internet Architecture: an Overview. Request for Comments RFC 1633, Internet Engineering Task Force, June 1994.

[BE89] R. Braden and Ed. Requirements for Internet Hosts - Communication Layers. Request for Comments RFC 1122, Internet Engineering Task Force, October 1989.

[BE95] F. Baker and Ed. Requirements for IP Version 4 Routers. Request for Comments RFC 1812, Internet Engineering Task Force, June 1995.

[BG98] Jill M. Boyce and Robert D. Gaglianello. Packet loss effects on MPEG video sent over the public internet. In *Proceedings of the sixth ACM international conference on Multimedia*, pages 181–190, 1998.

[BJS00] Lee Breslau, Sugih Jamin, and Scott Shenker. Comments on the performance of measurement-based admission control algorithms. In *Proceedings of the 2000 IEEE Computer and Communications Societies Conference on Computer Communications (INFOCOM-00)*, pages 1233–1242, Los Alamitos, March 26–30 2000. IEEE.

[BLFF96] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext Transfer Protocol – HTTP/1.0. Request for Comments RFC 1945, Internet Engineering Task Force, May 1996.

[BNkc02] Andre Broido, Evi Nemeth, and kc claffy. Internet expansion, refinement and churn. *European Transactions on Telecommunications*, 13(1):33–51, January 2002.

[BO97] L. Berger and T. O'Malley. RSVP Extensions for IPSEC Data Flows. Request for Comments RFC 2207, Internet Engineering Task Force, September 1997.

[Bol93] Jean-Chrysostome Bolot. Characterizing end-to-end packet delay and loss in the internet. *Journal of High-Speed Networks*, 2(3):305–323, December 1993.

[BS92] B.G. Barnett and E.T. Saulnier. High level traffic analysis of a LAN segment. In *17th Conference on Local Computer Networks, 1992. Proceedings.*, pages 188–197, September 1992.

[Cai] Internet tools taxonomy. On web, `http://www.caida.org/tools/taxonomy/`. Refered 2002-05-05.

[CBP95] K.C. Claffy, H.-W. Braun, and G.C. Polyzos. A parameterizable methodology for internet traffic flow profiling. *IEEE Journal on Selected Areas in Communications*, pages 1481–1494, October 1995.

[CC96a] Robert Carter and Mark Crovella. Dynamic server selection using bandwidth probing in wide-area networks. Technical Report 1996-007, Computer Science Department, Boston University, March 18 1996.

[CC96b] Robert L. Carter and Mark E. Crovella. Measuring bottleneck link speed in packet-switched networks. *Performance Evaluation*, 27&28:297–318, 1996.

[CCH95] Lily Cheng, SinMin Chang, and H. Hughes. A connection admission control algorithm based on empirical traffic measurements. In *1995 IEEE International Conference on Communications, 1995. ICC '95 Seattle, 'Gateway to Globalization'*, volume 2, pages 793–797, 1995.

[CCL99] Cooper Chang, Chung-Ju Chang, and Kuen-Rong Lo. Analysis of a hierarchical cellular system with reneging and dropping for waiting new and handoff calls. *IEEE Transactions on Vehicular Technology*, 48(4):1080–1091, July 1999.

[CDFT98] J. Callas, L. Donnerhacke, H. Finney, and R. Thayer. OpenPGP Message Format. Request for Comments RFC 2440, Internet Engineering Task Force, November 1998.

[CE97] K. Chandra and A.E. Eckberg. Traffic characteristics of on-line services. In *Second IEEE Symposium on Computers and Communications, 1997. Proceedings*, pages 17–21, 1997.

[CER99] Similar attacks using various rpc services. CERT® Incident Note IN-99-04, CERT Coordination Center, 1999. Available on `http://www.cert.org/incident_notes/IN-99-04.html`. Referred 2002-05-25.

[CJ99] Michele Clark and Kevin Jeffay. Application-level measurements of performance on the vBNS. In *ICMCS, Vol. 2*, pages 362–366, 1999.

[CM97] K. Claffy and T. Monk. What's next for internet data analysis? status and challenges facing the community. *Proceedings of the IEEE*, 85(10):1563–1571, October 1997.

[CMGR94] M. Cinotti, E.D. Mese, S. Giordano, and F. Russo. Long-range dependence in ethernet traffic offered to interconnected DQDB MANs. In *Singapore ICCS '94. Conference Proceedings*, volume 2, pages 479–484, November 1994.

[Com] Clientvantage – end-user experience monitoring. On web `http://www.compuware.com/products/vantage/clientvantage/`. Referred 2002-05-20.

[Com97] Technical Committee. LAN emulation over ATM version 2.0 — LUNI specification. Technical Report af-lane-0084.000, The ATM Forum, July 1997.

[Cor] Coralreef. On web, `http://www.caida.org/tools/-measurement/coralreef/`. Referred 2002-05-15.

[CPB93] K. C. Claffy, G. C. Polyzos, and H. W. Braun. Traffic characteristics of the T1 NSFNET backbone. *Proc. IEEE INFOCOM'93*, 2:885–892, 1993.

[Cri96] M. Crispin. Internet Message Access Protocol - Version 4rev1. Request for Comments RFC 2060, Internet Engineering Task Force, December 1996.

[DA99] T. Dierks and C. Allen. The TLS Protocol Version 1.0. Request for Comments RFC 2246, Internet Engineering Task Force, January 1999.

[DAG] The dag project. On web, `http://dag.cs.waikato.ac.nz`. Referred 2005-05-15.

[Dan99] Peter H. Dana. Global positioning system overview. On web, `http://www.colorado.edu/geography/gcraft/notes/gps/-gps_f.html`, 1999. Referred 2005-05-15.

[DNP99] M. Degermark, B. Nordgren, and S. Pink. IP Header Compression. Request for Comments RFC 2507, Internet Engineering Task Force, February 1999.

[Dow99] Allen B. Downey. Using pathchar to estimate internet link characteristics. In *SIGCOMM'99*, September 1999.

[Dra92] E. Drakopoulos. Analysis of a local computer network with workstations and x terminals. In *17th Conference on ocal Computer Networks, 1992. Proceedings.*, pages 206–215, September 1992.

[E.492] Service quality assessment for connection set-up and release delays. ITU-T Recommendation E.431, International Telecommunication Union, 1992.

[E.496] Grade of service (GOS) monitoring. ITU-T Recommendation E.493, International Telecommunication Union, 1996.

[E.497] Traffic measurement by destination. ITU-T Recommendation E.491, International Telecommunication Union, 1997.

[Ear02] Service agreement. On web, `http://www.earthlink.net/biz/-broadband/dedicated/agreement/`, 2002. Referred 2002-05-20.

[End] Irtf end-to-end intrest. Mailing list end-to-end@postel.org. Archives `http://www.postel.org/end-to-end/`.

[ENW96] A. Erramilli, O. Narayan, and W. Willinger. Experimental queueing analysis with long-range dependent packet traffic. *IEEE/ACM Transactions on Networking*, 4(2):209–223, April 1996.

[EW94] A. Erramilli and J.L. Wang. Monitoring packet traffic levels. In *IEEE Global Telecommunications Conference, 1994. GLOBECOM '94. Communications: The Global Bridge*, volume 1, pages 274–280, November 1994.

[Fel98] A. Feldmann. Continuous online extraction of http traces from packet traces, 1998.

[FGHW99] Anja Feldmann, Anna C. Gilbert, Polly Huang, and Walter Willinger. Dynamics of IP traffic: A study of the role of variability and the impact of control. In *SIGCOMM'99*, September 1999.

[FGM+99] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. Request for Comments RFC 2616, Internet Engineering Task Force, June 1999.

[FH98] Paul Ferguson and Geoff Huston. *Quality of Service: delivering QoS on the Internet and in corporate networks.* John Wiley & Sons, 1998.

[FS00] P. Ferguson and D. Senie. Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. Request for Comments RFC 2827, Internet Engineering Task Force, May 2000.

[G.100] The e-model, a computational model for use in transmission planning. ITU-T Recommendation G.107, International Telecommunication Union, 2000.

[GD98] Ian Graham and Stephen Donnelly. Comparative measurement of QoS on the trans-pacific internet. In Raif O. Onvural, Seyhan Civanlar, Paul J. Doolan, and James V. Luciani, editors, *Proc SPIE–The International Society for Optical Engineering: Internet Routing and Quality of Service v 3529*, pages 289–294, Boston, MA, USA, 1998.

[GSC+96] Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. Request for Comments RFC 1889, Internet Engineering Task Force, January 1996.

[GVE00] A. Gulbrandsen, P. Vixie, and L. Esibov. A DNS RR for specifying the location of services (DNS SRV). Request for Comments RFC 2782, Internet Engineering Task Force, February 2000.

[HFGC98] D. Hoffman, G. Fernando, V. Goyal, and M. Civanlar. RTP Payload Format for MPEG1/MPEG2 Video. Request for Comments RFC 2250, Internet Engineering Task Force, January 1998.

[Hus98] Geoff Huston. *ISP Survival Guide: strategies for running a competitive ISP*. John Wiley & Sons, Inc, 1998.

[I.393] General aspects of quality of service and network performance in digital networks, including isdns. ITU-T Recommendation I.350, International Telecommunication Union, 1993.

[IPF02] IP flow information export (ipfix) charter. On web, `http://www.ietf.org/html.charters/ipfix-charter.html`, 2002. referred 2002-05-05.

[ITU99] ITU-T. Internet protocol data communication service - IP packet transfer and availability performance parameters. ITU-T Recommendation I.380, International Telecommunication Union, February 1999.

[Jac88] V. Jacobson. Congestion avoidance and control. In *Proceedings of the ACM SIGCOMM Conference*, pages 314–329, August 1988.

[Jac90] V. Jacobson. Compressing TCP/IP headers for low-speed serial links. Request for Comments RFC 1144, Internet Engineering Task Force, February 1990.

[Jac97] Van Jacobson. Pathchar: How to infer the characteristics of internet paths. Lecture at Mathematical Sciences Research Institute, April 1997.

[JBB92] V. Jacobson, R. Braden, and D. Borman. TCP Extensions for High Performance. Request for Comments RFC 1323, Internet Engineering Task Force, May 1992.

[KH02] Jorma Jormakka and Kari Heikkinen. QoS/GOS parameter definitions and measurement in IP/ATM networks. In *Proceedings of First COST 263 International Workshop, QofIS 2000*, pages 182–193, Berlin, Germany, September 2002.

[Joh93] M. St. Johns. Identification Protocol. Request for Comments RFC 1413, Internet Engineering Task Force, February 1993.

[Jor94] J. Jormakka. A model for service switching point for dimensioning of intelligent networks. In *Intelligent Network '94 Workshop*, pages 1227–1243, May 1994.

[JR96] Raj Jain and Shawn A Routhier. Packet trains - measurements and a new model for computer network traffic. *IEEE Journal on Selected Areas in Communications*, 4(6):986–995, September 1996.

[JW97] J.L. Jerkins and J.L. Wang. A measurement analysis of ATM cell-level aggregate traffic. In *IEEE Global Telecommunications Conference, 1997. GLOBECOM '97.*, volume 3, pages 1589–1595, November 1997.

[KBC97] H. Krawczyk, M. Bellare, and R. Canetti. HMAC: Keyed-Hashing for Message Authentication. Request for Comments RFC 2104, Internet Engineering Task Force, February 1997.

[Kiv94] Kaj Kivinen. Milestones in telecommunications in some selected countries. Report 3/94, Laboratory of Telecommunications Technology at Helsinkin University of Technology, 1994.

[KL86] B. Kantor and P. Lapsley. Network News Transfer Protocol. Request for Comments RFC 977, Internet Engineering Task Force, February 1986.

[KN74] L. Kleinrock and W. E. Naylor. On measured behavior of the ARPA network. *AFIPS*, 43:767–780, 1974.

[KqL96] Yonghwan Kim and San qi Li. Timescale of interest in traffic measurement for link bandwidth and allocation design. In *Proceedings IEEE INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation.*, volume 2, pages 738–748, 1996.

[LAJ98] C. Labovitz, A. Ahuja, and F. Jahanian. Experimental study of internet stability and wide-area backbone failures. Technical Report CSE-TR-382-98, University of Michigan Department of Electrical Engineering and Computer Science, December 16, 1998.

[Lam95] M. Lambert. A Model for Common Operational Statistics. Request for Comments RFC 1857, Internet Engineering Task Force, October 1995.

[LFJ97] Beng Ong Lee, Victor S. Frost, and Roelof Jonkman. Netspec 3.0 source models for telnet, ftp, voice, video and www traffic. Technical Report ITTC-TR-10980-19, Information & Telecommunication Technology Center, The University of Kansas, 2291 Irving Hill Road Lawrence, KS 66045, January 1997.

[LIP99] Marko Luoma, Mika Ilvesmäki, and Markus Peuhkuri. Source characteristics for traffic classification in differentiated services type of networks. In *Proceedings of Voice, Video and Data '99*. SPIE, September 1999.

[LM98] Qiong Li and D.L. Mills. On the long-range dependence of packet round-trip delays in internet. In *1998 IEEE International Conference on Communications, 1998. ICC 98. Conference Record*, volume 2, pages 1185–1191, June 1998.

[Lof97] Siegfried Loffer. Using flows for analysis ona measurement of internet traffic. Master's thesis, Institute of Communication Networks and Computer Engineering (IND) of the University of Stuttgart, August 1997.

[LP99] Anna-Kaisa Lindfors and Markus Peuhkuri. Vulnerabilities of ftp protocol, ftp servers and clients. In *Tik-110.452 Tietojärjestelmien käytännön turvallisuuden erikoiskurssi*. Helsinki University of Technology. Telecommunications Software and Multimedia Laboratory, 1999. Also available from `http://www.iki.fi/puhuri/htyo/Tik-110.452/`.

[LTWW93] Will E. Leland, Murad S. Taqq, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of Ethernet traffic. In Deepinder P. Sidhu, editor, *ACM SIGCOMM*, pages 183–193, San Francisco, California, 1993.

[LTWW94] Will E. Leland, Murad S Taqqu, Walter Willinger, and Dalinel V Wilson. On the self-similar nature of ethernet traffic. *IEEE/ACM Transactions on Networking*, 2(1), 1994.

[MA01] M. Mathis and M. Allman. A Framework for Defining Empirical Bulk Transfer Capacity Metrics. Request for Comments RFC 3148, Internet Engineering Task Force, July 2001.

[Mah97] B.A. Mah. An empirical model of http network traffic. In *Proceedings IEEE INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution*, volume 2, pages 592–600, 1997.

[MD90] J.C. Mogul and S.E. Deering. Path MTU discovery. Request for Comments RFC 1191, Internet Engineering Task Force, November 1990.

[Mea] Preliminary measurement spec for internet routers. Draft at `http://www.caida.org/tools/measurement/measurementspec/`. Work in process.

[MH00] A. Mena and J. Heidemann. An empirical study of real audio traffic. In *Proceedings of the IEEE Infocom '00*, pages 101–110, March 2000.

[MIIA99] O. Maeshima, Y. Ito, M. Ishikura, and T. Asami. A method of service quality estimation with a network measurement tool. In *1999 IEEE International Performance, Computing and Communications Conference*, pages 201–209, February 1999.

[Mil92] David L. Mills. Network Time Protocol (Version 3) Specification, Implementation. Request for Comments RFC 1305, Internet Engineering Task Force, March 1992.

[ML97] N.F. Maxemchuk and S. Lo. Measurement and interpretation of voice traffic on the internet. In *1997 IEEE International Conference on Communications, 1997. ICC '97 Montreal, Towards the Knowledge Millennium*, volume 1, pages 500–507, June 1997.

[MMFR96] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow. TCP Selective Acknowledgement Options. Request for Comments RFC 2018, Internet Engineering Task Force, October 1996.

[Moc87] P.V. Mockapetris. Domain names - concepts and facilities. Request for Comments RFC 1034, Internet Engineering Task Force, November 1987.

[Mog92] Jeffrey C. Mogul. Observing TCP dynamics in real networks. WRL Research Report 92/2, Digital Western Research Laboratory, April 1992.

[Moo02] K. Moore. On the use of HTTP as a Substrate. Request for Comments RFC 3205, Internet Engineering Task Force, February 2002.

[MP99] J. Mahdavi and V. Paxson. IPPM Metrics for Measuring Connectivity. Request for Comments RFC 2678, Internet Engineering Task Force, September 1999.

[MR96] J. Myers and M. Rose. Post Office Protocol - Version 3. Request for Comments RFC 1939, Internet Engineering Task Force, May 1996.

[MSMO97] M. Mathis, J. Semke, J. Mahdavi, and T. Ott. Macroscopic behavior of the TCP congestion avoidance algorithm. *Computer Communication Review*, 27(3):67–82, July 1997.

[Muu] Mike Muuss. The story of the TTCP program. On web, `http://ftp.arl.mil/~mike/ttcp.html`.

[Neta] Cisco IOS netflow. On web `http://www.cisco.com/warp/public/732/Tech/netflow/`. Referred 2002-05-15.

[NeTb] Netramet. On web `http://www2.auckland.ac.nz/net/-Accounting/ntm.Release.note.html`. Referred 2002-05-15.

[Net95] Netperf: A network performance benchmark, revision 2.0, February 1995.

[NGBS+97] Henrik Frystyk Nielsen, Jim Gettys, Anselm Baird-Smith, Eric Prud'hommeaux, Håkon Lie, and Chris Lilley. Network performance effects of HTTP/1.1, CSS1, and PNG. In *Proceedings of the ACM SIGCOMM Conference : Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM-97)*, volume 27,4 of *Computer Communication Review*, pages 155–166, New York, September 14–18 1997. ACM Press.

[Nie93] Jakob Nielsen. *Usability Engineering*. AP Professional, Boston, 1993.

[Nie97] Jakob Nielsen. The need for speed. Technical report, Usable Information Technology, March 1997.

[Nie99] Jakob Nielsen. The top ten new mistakes of web design. Alertbox column, useit.com, May 1999.

[Nok] Nokia DX 200 fin6 user documentation.

[PAMM98] V. Paxson, G. Almes, J. Mahdavi, and M. Mathis. Framework for IP Performance Metrics. Request for Comments RFC 2330, Internet Engineering Task Force, May 1998.

[Pax94] V. Paxson. Empirically derived analytic models of wide-area TCP connections. *IEEE/ACM Transactions on Networking*, 2(4):316–336, 1994.

[Pax96] V. Paxson. End-to-end routing behavior in the internet. In *Proceedings of the ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, volume 26,4 of *ACM SIGCOMM Computer Communication Review*, pages 25–39, New York, August 26–30 1996. ACM Press.

[Pax97a] Vern Paxson. Automated packet trace analysis of TCP implementations. In *Proceedings of the ACM SIGCOMM Conference : Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM-97)*, volume 27,4 of *Computer Communication Review*, pages 167–180, New York, September 14–18 1997. ACM Press.

[Pax97b] Vern Paxson. End-to-end Internet packet dynamics. In *Proceedings of the ACM SIGCOMM Conference : Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM-97)*, volume 27,4 of *Computer Communication Review*, pages 139–154, New York, September 14–18 1997. ACM Press.

[Pax97c] Vern E. Paxson. *Measurements and Analysis of End-to-End Internet Dynamics*. Technical report, University of California, Berkeley, April 1997.

[Pax98] Vern Paxson. On calibrating measurements of packet transit times. Technical Report LBNL-41535, Lawrance Berkeley National Laboratory, Network Researh Group, 1998.

[Peu97] Markus Peuhkuri. Prosessien vuorottelun vaikutus dataliikenteeseen. Master's thesis, Helsinki University of Technology, May 1997.

[Peu01] Markus Peuhkuri. A method to compress and anonymize packet traces. In *Proceedings of the First ACM SIGCOMM Internet Measurement Workshop*, pages 257–260, San Francisco, USA, November 2001. ACM SIGCOMM.

[PF95] Vern Paxson and Sally Floyd. Wide area traffic: the failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, June 1995.

[PMAM98] V. Paxson, J. Mahdavi, A. Adams, and M. Mathis. An architecture for large scale internet measurement. *IEEE Communications Magazine*, 36(8):48–54, August 1998.

[Pos80] J. Postel. User Datagram Protocol. Request for Comments RFC 768, Internet Engineering Task Force, August 1980.

[Pos81a] J. Postel. Internet Control Message Protocol. Request for Comments RFC 792, Internet Engineering Task Force, September 1981.

[Pos81b] J. Postel. Internet Protocol. Request for Comments RFC 791, Internet Engineering Task Force, September 1981.

[Pos81c] J. Postel. Transmission Control Protocol. Request for Comments RFC 793, Internet Engineering Task Force, September 1981.

[PR83] J. Postel and J.K. Reynolds. Telnet Protocol Specification. Request for Comments RFC 854, Internet Engineering Task Force, May 1983.

[PR85] J. Postel and J.K. Reynolds. File Transfer Protocol. Request for Comments RFC 959, Internet Engineering Task Force, October 1985.

[PS89] C. Pattinson and R.M. Strachan. The characterisation of network applications-parameters for artificial traffic generation. In *Sixth United Kingdom Teletraffic Symposium, 6th.*, pages 2/1–2/6, 1989.

[Rah01] K. Rahko. History of telephone traffic measurements. Email discussion with author, September 2001.

[RFB01] K. Ramakrishnan, S. Floyd, and D. Black. The Addition of Explicit Congestion Notification (ECN) to IP. Request for Comments RFC 3168, Internet Engineering Task Force, September 2001.

[Riv92] R. Rivest. The MD5 Message-Digest Algorithm. Request for Comments RFC 1321, Internet Engineering Task Force, April 1992.

[RL95] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4). Request for Comments RFC 1771, Internet Engineering Task Force, March 1995.

[Ros95] O. Rose. Statistical properties of MPEG video traffic and their impact on traffic modeling in ATM systems. In IEEE Computer Society. Technical Committee on Computer Communications, editor, *Proceedings: 20th Conference on Local Computer Networks, October 16–19, 1995, Minneapolis, Minnesota*, volume 20, pages 397–406, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1995. IEEE Computer Society Press.

[Sch96] Bruce Schneier. *Applied Cryptography Second Edition : protocols, algorithms, and source code in C.* John Wiley & Sons, Inc., 1996.

[SPI97] *DBS: A Powerful Tool for TCP Performance Evaluations*, SPIE Proceedings of Performance and Control of Network Systems, November 1997.

[SPS+01] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Stephen T. Kent, and W. Timothy Strayer. Hash-Based IP traceback. In Roch Guerin, editor, *Proceedings of the ACM SIGCOMM 2001 Conference (SIGCOMM-01)*, volume 31, 4 of *Computer Communication Review*, pages 3–14, New York, August 27–31 2001. ACM Press.

[SPW] Stuart Staniford, Vern Paxson, and Nicholas Weaver. How to 0wn the internet in your spare time. In *Proceedings of the 11th USENIX Security Symposium (Security '02).* To appear.

[SSZ+96] A. Srikitja, M.A. Stover, T. Zhang, E. Zhong, S. Banerjee, D. Tipper, M.B. Weiss, and A. Khalil. Analysis of traffic measurements on a wide area ATM network. In *Global Telecommunications Conference, 1996. GLOBECOM '96. 'Communications: The Key to Global Prosperity*, volume 1, pages 778–782, November 1996.

[Stu01] Rick Sturm. SLA metrics. *Network World Network Systems Management Newsletter*, October 2001. On web, `http://www.nwfusion.com/newsletters/nsm/2001/01083536.html`.

[SW98] M. Siler and J. Walrand. On-line measurement of QoS for call admission control. In *1998 Sixth International Workshop on Quality of Service, 1998. (IWQoS 98).*, pages 39–48, May 1998.

[SWKA00] Stefan Savage, David Wetherall, Anna Karlin, and Tom Anderson. Practical network support for IP traceback. In *Proceedings of the 2000 ACM SIGCOMM Conference*, August 2000. An early version of the paper appeared as techreport UW-CSE-00-02-01 available at: `http://www.cs.washington.edu/homes/savage/traceback.html`.

[TDG98] R. Thayer, N. Doraswamy, and R. Glenn. IP Security Document Roadmap. Request for Comments RFC 2411, Internet Engineering Task Force, November 1998.

[TMW97] K. Thompson, G.J. Miller, and R. Wilder. Wide-area internet traffic patterns and characteristics. *IEEE Network*, 11(6):10–23, November 1997.

[Tre00] About the PSC treno server. On web, `http://www.psc.edu/networking/treno_info.html`, July 2000. referred 2002-05-05.

[Wan01] Zheng Wang. *Internet QoS: architechtures and mechanisms for Quality of Service*. Academic Press, 2001.

[WHP99] B. Wijnen, D. Harrington, and R. Presuhn. An Architecture for Describing SNMP Management Frameworks. Request for Comments RFC 2571, Internet Engineering Task Force, April 1999.

[Wii01] Peter Wiilis, editor. *Carrier-scale IP networks: designing and operating Internet networks*. Number 1 in BT Communications Technology series. BT Exact Technologies, 2001.

[Y.100] Relationships among ISDN, Internet protocol, and GII performance recommendations. ITU-T Recommendation Y.1501, International Telecommunication Union, 2000.

[Y.102] Network performance objectives for IP-based services. ITU-T Recommendation Y.1541, International Telecommunication Union, 2002. Pre-published.

List of Tables

List of Figures