Information Society Technologies

MOME
Monitoring and Measurement Cluster – 001990

D21: MOME Database

Abstract

Deliverable D21 documents the design process of the MOME Database, which will store the information about raw measurement data repositories.

The design process was started with a study of the state of the art, which showed that most projects with similar goals have opted for a meta-database model which provides links to the original data sources as well as the additional value of centralised, documented access to them.

The design process was performed in several steps:

- Use case study
- Design of the object models and database tables
- Hardware and Software Requirements analysis

To validate the database, a prototype implementation was deployed in the premises of Salzburg Research. This prototype will serve as the basis for the Web based database access interface which will be deployed at the TERENA data centre. This deployment will be documented in Deliverable D22.

Keywords

Deliverable D21, MOME Database, Meta-database, State of the Art, Design

Document Info	
Document Reference	*MOME-WP2-0409_D21_MOME_DATABASE*
Document Type	Deliverable
Deliverable Type	Report
Deliverable Status	Submitted
Deliverable Date	Contractual: 30/09/2004, Actual: 30/09/2004
Dissemination Level	Public
Editing Author	*Pedro A. Aranda Gutiérrez, TID*
Contributing Author(s)	*Marek Dabrowski, WUT* *Attila Vidacs, BUTE* *Kardos Sandor Zsolt, BUTE* *Felix Strohmeier, SRF* *Carsten Schmoll, FHG*
Workpackage(s)	WP2

Table of Contents

List of Figures

List of Tables

Executive Summary

This document describes the MOME database, the process followed by WP2 for it's definition and the deployment of the first prototype.

Chapter 1 provides an introduction to the document.

Chapter 2 presents a survey of the state of the art in measurement projects and measurement repositories in the Internet. This study shows that most of the projects which coordinate measurements in repositories use a meta-database approach to implement a measurement catalogue. The database as such doesn't contain the measurements proper, but the information which is necessary to get access to the database.

Chapter 3 presents the database design process, which is based on a use case analysis. This yields the definition of the tables describing the database objects and their relationships.

Chapter 4 includes the DBMS HW and SW requirement analysis.

Chapter 5 describes the prototype implementation, which took place at the venues of Salzburg Research.

Two annexes are included in this document which include an overview of existing Internet measurement repositories and an overview of analysis tools.

List of Acronyms

ATM	Asynchronous Transfer Mode
BGP-4	Border Gateway Protocol
CAIDA	Cooperative Association for Internet Data Analysis
CVS	Concurrent Version System
DoS	Denial of Service
EGSO	European Grid of Solar Observations
FTP	File Transfer Protocol
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
IS-IS	Intermediate System to Intermediate System
LAN	Local Area Network
MTRD	Multithreaded Routing Daemon
NASA	National Aeronautics and Space Agency
OWD	One Way Delay
POS	Packet over SONET
QoS	Quality of Service
RDBMS	Relational Database Management System
RTT	Round Trip Time
SIMR	Scalable Internet Measurement Repository
URL	Universal Resource Locator
VLAN	Virtual Local Area Network
WAN	Wide Area Network
WLAN	Wireless Local Area Network
WWW	World Wide Web

1 Introduction

The main objective of the MOME project is the coordination of several European activities or projects in the field of traffic measurement and monitoring by offering a platform for knowledge, tool and data exchange. Therefore MOME project site intends to be the meeting place for researchers in the field of Internet Measurement and Monitoring and an information exchange on on-going measurements performed by different projects.

The purpose of this deliverable is the description of the design process of the MOME database. Currently open projects where Internet measurement databases are being created or logically interconnected were taken into account.

While gathering information about projects and tools related to the creation of similar databases to the one that this document describes, an important project outside the realm of Internet measurement and monitoring was identified. This project, called EGSO, "European Grid of Solar Observations" is creating an overlay over networked solar observations. The final objective of the project is to develop a metadata model for this type of data. The way which similar projects have followed is the use of a flexible meta-database architecture too. Some organisations such as CAIDA, "Cooperative Association for Internet Data Analysis" and projects such as Mark Allman's SIMR ("Scalable Internet Measurement Repository") architecture are trying to create models of meta-databases of measurements. In our opinion, the meta-database architecture is the best way to achieve the goal of creating an access overlay to the measurements gathered.

During the definition phase of the MOME project a database to store measurement traces was intended. The state of the art showed, that an access overlay to the measurements of the associated projects was more desirable, since most of the measurement projects post their data on their own project sites. This overlay, called meta-database, stores measurement descriptions and pointers to the actual measurement data in the form of the URL which has to be used to retrieve the measurement data from the project web site.

The design process was then divided into several separate steps:

- Use case study: Both human interaction with the database and automated processes access to the database initiated by a computer are covered in this study.

- Design of the object models and database tables. Tables include common attributes and a set of tables storing additional attributes and results of analysis tasks applicable to the different types of traces. These types include packet traces, QoS measurements, routing data, HTTP traces. Web-based data repositories are also taken into account. A separate table stores the user attributes for registered MOME users.

- Hardware and Software Requirements analysis for the deployment of the database.

Finally, this deliverable also includes a short description of the prototype machine on which the design has been tested. In addition, Annex A shows a survey of the existing data repositories and Annex B explains the state of the art in data analysis.

2 State of the art

Nowadays, cooperation between Internet Providers in order to use routing information data is very low. Some factors are important to understand this issue. On the one hand, competitive providers

invest huge amounts of money to increase networks capacity instead of investing to know how that capacity is utilised and trying to improve that use. This would be a more engineering-oriented solution but more resources and time would be needed. On the other hand, there is a lack of tools, identified parameters and information data to be able to compare traffic results of an independent provider to manage network growth. This lack of cooperation matches in time with the growth in the number of applications using distributed data and computing resources throughout the Internet.

The state of the art analysis features two approaches to meta-database designs which are applicable in the MOME database context and is complemented by a survey of measurement and traces repositories available in the Internet.

During the study of the state of the art in Internet Measurement Databases, the IST project "European Grid of Solar Observations" EGSO [1] was identified. EGSO is an IST/EUROGRID project. The consortium is formed by 10 institutions which include astrophysics and computer science experts. The objective of the project is to develop a metadata model for Solar Observations and use it in a GRID based Database of Solar Observations. Their work is well into the second year and at this point they have agreed on a standard metadata representation for solar observations. A basic Web (PHP) based interface is operative. The project is closely collaborating with other international projects in the field of Solar Observation like the US Virtual Solar Observatory, which is funded by NASA.

2.1 Meta-database models

2.1.1 CAIDA: Internet Measurement Data Catalogue (IMDC)

The main objective of the project "Correlating Heterogeneous Measurement Data to Achieve System-level Analysis of Internet Traffic Trends" is to foster the progress of measurement-based network research. In order to achieve this goal, it is important to perform the integration of several technologies. This leads to the creation of a metadata repository that facilitates the access to the distributed raw data repositories. This database, known as the Internet Measurement Data catalogue (IMDC), will assist to the development of reports and new tools concerning "well captured" traffic, such as analysis and correlated visualization tools. Furthermore, a new language for labelling and annotating data sets will be created.

The proposed metadata repository will allow researches to investigate some aspects such as the level of fragmented traffic, encrypted traffic, traffic favouritism, path symmetry, address space utilization and consumption, directional balance of traffic volume, routing protocol behaviour and policy, distribution statistics of path lengths, flow sizes, packet sizes, prefix lengths, and routing announcements. In addition, bringing together researchers and developers will be accomplished. At the present time, CAIDA is working at the creation of this central database.

Only metadata (i.e. "data about data") is contained in the central database. In the first draft of the design, the database consists of a series of actors, files, collections, packages and annotations.

Actors who deal with tools and data, are creators, archivers and contributors. Creators create a certain tool or piece of data, archivers store in order a given tool or piece of data and contributors catalogue a given tool or set of data. Some users of the catalogue are authors, searchers, analysts and annotators. Authors make a study using data stored in the database. Searchers make queries to the database. Analysts do research from metadata and annotators annotate items in the database.

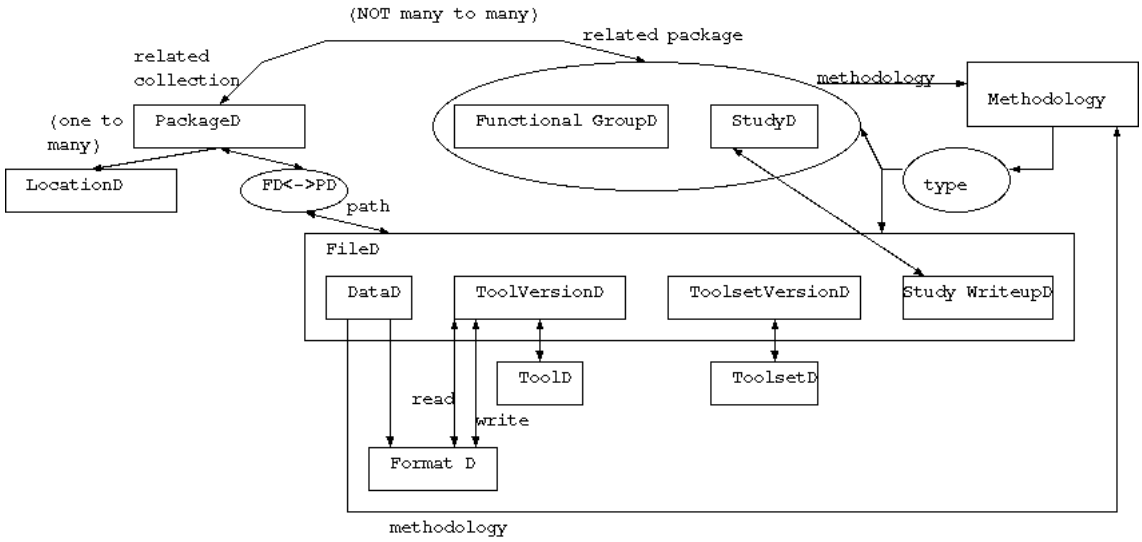

Figure 2-1: Initial CAIDA measurement meta-database design

The figure above shows the database structure first draft of the original design. File types within the database are data, tools, toolsets and study write-ups. Several common data formats will be created depending on the tool version. Tools and toolsets allow users to compare and monitor different traffic data. On a physical level, files are grouped into Packages which are "downloadable" units and can be situated at multiple locations (i.e. mirrors). On a logical level, files can be organized into collections which can be either functional groups or studies. Studies write-ups can be created and archived in the database. Collections are related to packages in a not many to many way. Methodologies determine which data parameters and attributes are vital to network management and consequently which are the actions of the database.

One important design concept is annotations. Comments, tool-related bugs and data format definition belong to annotations field. Annotations will provide necessary flexibility and extensibility to the database.

Entries in the database must have as mandatory fields the entry date and the contributor. Other optional fields are supported, such as annotations, administrative information and update date.

2.1.2 SIMR architecture

In March 2002, Mark Allman proposed an architecture called "Scalable Internet Measurement Repository" (SIMR) [2]. This architecture has three main components:

1. **Measurement Repositories (MRs)**: Locations around the Internet having measurement results available

2. **Clients**: representing researchers using the measurement data with their web-browsers

3. **Database:** The centralised key component of the architecture

In addition to the approach with the centralised measurement database, an alternative approach where clients directly communicate with the measurement repositories is discussed. The basic design of SIMR has similarities with music sharing systems like Napster, or, in the alternative approach, with directly communicating peer-to-peer data exchange systems. SIMR is separated into five databases, most of them specified by a single table, with relations in between. The following figure shows the rough design and the interconnections of the database, which only contains meta-info about measurements. The measurement results itself are linked by the URL to the original measurement data source either by HTTP or FTP. Therefore the measurement data must be a single directly linked file, an MD5 hash of the data is kept in the meta-database for validation.

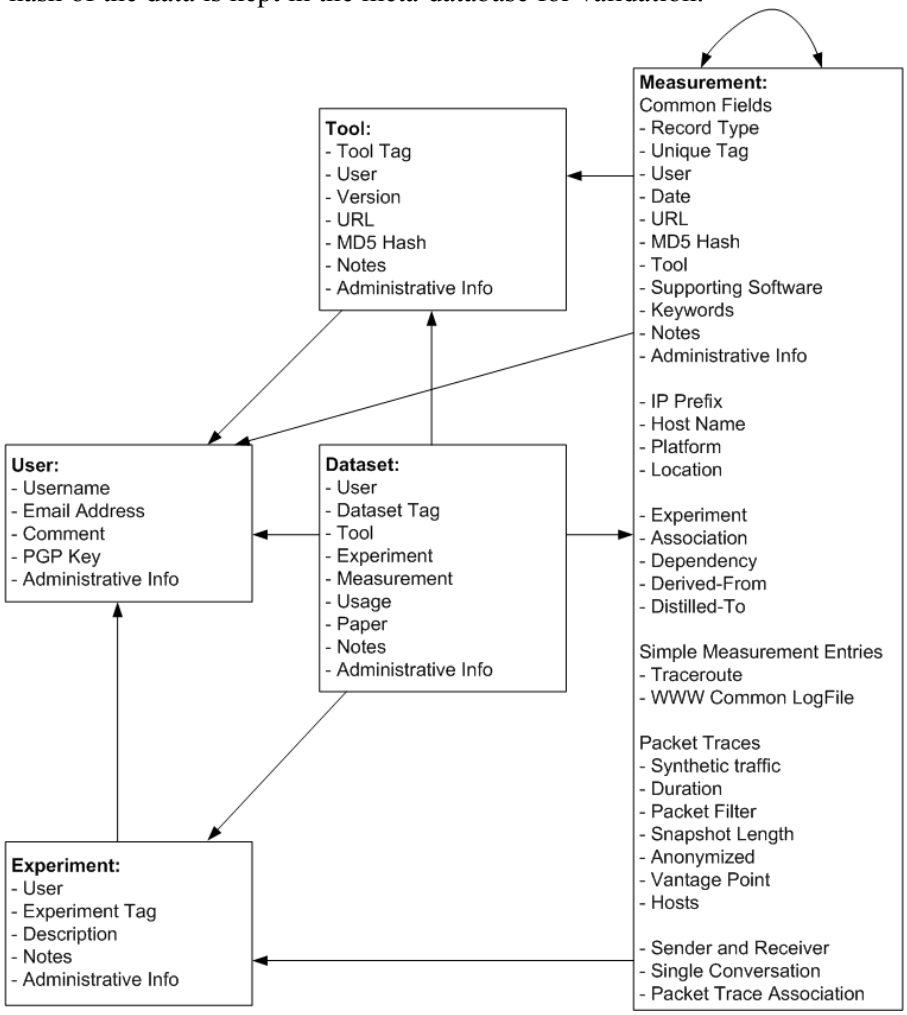

Figure 2-2: SIMR database structure

In his paper, Allman also addresses important points like security considerations and strategies against database pollution (e.g. metadata validation and URL verification). SIMR bases on submissions from known researchers, which has the advantage, that it prevents databases pollution/DoS attacks, but makes the system depending on these researchers.

2.1.3 Sprint IP Monitoring Project

Motivation:

The Sprint IP Monitoring Project (http://ipmon.sprint.com/) collects very large sets of detailed packet-level data from a tier-1 backbone network. A comprehensive framework is needed for efficiently managing the metadata and ultimately the data itself within the project.

Traces data:

The traces consist the following:

- packet traces of different links (50-100 GBytes)
- configuration information (topology, etc.)
- BGP-4 routing tables (downloaded from routers)
- IS-IS contingency tables (downloaded from routers)

Problems and experiences:

- The total amount of data is on the order of tens of terabytes, thus on-line storage is not possible.
- Facility of sharing the result datasets is needed.
- Different types of data need to be correlated in a systematic way. (E.g., raw data traces mapped with associated routing tables and topology information)
- Given the need to reuse results, a way is needed to determine which datasets are affected by a bug discovered in a piece of analysis software.
- Informal contacts are inadequate if more than 10 people are using the data.

Goal:

To design a system for **managing the metadata relating to traces and analysis results**, and **storage management**.

The problem breaks down into three areas:

- **storage of data** (traces, tables, results);
  [Storage management is done in an ad-hoc way and “they believe it is best left this way”, because the datasets are too large to manage in a conventional database.]
- **source code maintenance** of analysis programs;
  [Source code management is performed well by systems such as CVS and “they not intend to reinvent the wheel”.]
- **metadata management.**
  [see below]

Metadata Abstraction:

The proposed system is based on four key abstractions of the problem:

1. **Raw input data sets**;
   This includes the trace files themselves, but also additional data relating to the network, in particular BGP routing tables in effect when the trace was acquired, and information about the topology. Each dataset has a varying number of attributes (e.g., when the trace was taken, its duration, monitoring point, trace data format). Input data sets never change.

2. **Analysis programs;**
   Each *version* of a program used in the project is represented by an entity in the system. All these reside in a version control repository. The motivation for representing each separate version of a piece of software as metadata is this: the software inevitably evolves, functionality is added, formats change, bugs are fixed. Results obtained are potentially tied to a version of the software that produced it.

3. **Result data sets;**
   The difference between results datasets and raw input data sets is than (in principle!) it can be regenerated from the original data.

4. **Analysis operation;**
   An analysis operation is a combination of input datasets and programs, which generates a number of result datasets as output. For each operation, we need to keep track of the time, input datasets, specific version of programs, output datasets, and precisely what was done to produce them. (E.g., we might record the Unix command line used to process the data.)

Metadata model:

The four kinds of abstraction above are time-invariant. They naturally form a **dependency graph** , where the arcs are analysis operations and the nodes are datasets and program versions.

Design and implementation:

The dependency graph for the project metadata is easily represented in a relational database schema, and consequently stored in a RDBMS (in their case PostgreSQL).

Interaction with version control.

the database can interface to the version control repository by referencing modules and major release numbers (e.g., using CVS's version tagging capabilities).

Linkage to data storage system.

Trace files are identified by canonical file names, which encode where and when the trace was taken (with the format used by CAIDA). It also makes the metadata independent of file location.

User interfaces.

There are two user interfaces to the metadata, a GUI and a command-line interface.

Incremental deployment and enhancements.

An important advantage of implementing the metadata store as a database is the ability to deploy new functionality incrementally.

2.2 Data repositories

The survey of public data repositories was performed in order to get knowledge about actually available raw data sets and the way they are described by additional metadata. Detailed results of the survey can be found in Annex A. It should be noted, that quite a large amount of data is now available in the Internet. In many cases, these data are freely accessible. However, the data are stored in many independent, uncoordinated repositories, in different, incompatible formats.

2.3 Data analysis

The users of the MOME trace database should be provided with additional statistical information, which helps to initially evaluate the contents of particular raw data set. To support the design of the MOME database, the survey of state of the art in possible tasks and methods of data analysis was performed (detailed results can be found in Annex B).

One can point on different aspects of measurement data analysis. This is related to the fact that the range of possible uses of measurement data is very broad. The identified data analysis tasks include:

- pre-processing of data
- calculating empirical statistical parameters and distributions
- fitting to probability distributions
- fitting parameters of traffic models
- estimation of self-similarity parameters
- traffic prediction
- assessment of required bandwidth
- assessment of traffic descriptor parameters
- traffic matrix estimation
- QoS analysis

and

- intrusion detection methods.

The selection of the most important analysis tasks is reflected in the design of metadata tables for particular types of raw data.

2.4 Conclusions

The study of the state of the art has shown that the main problematic of coordinating Internet measurements is the diversity in the nature of said measurements. This is due to the myriad of tools which exist and which have been proven to be relevant by their day to day use.

This study has also revealed that there are projects trying to establish an umbrella to correlate Internet measurements. The perfect candidate for collaboration and synchronisation of database formats

would be the project initiated by CAIDA, which is in a very early stage and hasn't produced any database definition which MOME could reuse.

In general, the objective of establishing a database of measurement databases is a very ambitious objective, as shown by the number of partners and resources invested in the Solar Observation Database of the EGSO project. This project is showing first results now that they are well into their second year.

The common denominator of all projects studied is the use of metadata. The projects themselves don't aim at storing locally the traces they reference, but instead to provide value added data which include pointers to publicly available traces. Locally stored traces can be made available by URL's which point to an HTTP or FTP file server collocated with the main metadata database. This is the approach followed by the MOME project.

3 Database design

Research performed with the aid of measurement data requires a detailed documentation of the environment where the measurements were taken, including among others the type of measured network, level of traffic aggregation, location of measurement equipment. This is desirable e.g. for assessing the representativeness of the used traces and their relevance for particular research targets. Including in the MOME data-base carefully selected metadata, which annotate the actual raw data sets, will help the researchers to obtain this important additional information.

The database design process was split into use case study and the database table design proper. The use case study covers human interaction with the database as well as the cases where automated processes initiated by a computer access the database.

The MOME database will interact with different kind of actors:

- Unregistered Web users: Web users with no relationship to MOME or it's associated projects, which browse the MOME database
- Registered Web users: People interested in the work of MOME and/or it's associated projects, which contribute to the database and use it to retrieve measurements from other projects
- System Manager: Keeps the MOME database alive, by doing periodical checks and operations on it. Can be triggered by both, human interaction and autonomous processes.

3.1 Use cases

The use case study is based on a high level description of the full functionality needed to implement a working database with all kinds of interesting functionalities. The initial intention is to implement a minimal subset rendering a usable database and enhance it during the projects lifetime and beyond, depending on available workforce.

3.1.1 Interaction with Unregistered Web Users

This section lists what a visitor of the database shall be capable of doing with the system:

3.1.1.1 Get Introductory Information

The Web interface will provide a start-up screen, some static introductory Web pages introducing the server, explaining what the service is good for and optionally showing some use some examples, e.g. links to lists, a search result, and a detailed description page for one trace. A FAQ on the MOME database should be available.

3.1.1.2 Browse Database (all/by category)

This function enables the user to get a quick overview about what's inside the database. It will provide an overview of the database contents in a tabular format (html table). No user input is required, the selection of a category is optional. The columns will contain the main attributes of the database contents, like name, submitter, file size, measurement date, submission date and a short description, which are common to all kinds of traces. Columns may be selectable by the user. The interface should be sortable by column. The entry must be clickable to guide the user to the detail page for the data entry, which should show:

- 'Details' table with further detailed data description. Columns or attributes will depend on the type of described data, i.e. if it is a packet trace, flow trace, QoS results set, or routing data, etc.

- Results of analyses, if available.

3.1.1.3 Search Database

This function allows the user to input values for the most used attributes such as raw data size, originator, date, or even some of the statistics values and submit a match all/match any search to the database. The attributes on which filtering can be applied are depending on the data type.

3.1.1.4 Download Raw Data

This function allows the user to retrieve raw data from the URI specified in MOME database. The URI may point to data local to the MOME database or to any location in the Internet. In order to download raw data, the user doesn't need to be registered. In case an unregistered user accesses raw data, a message encouraging registration should be generated.

3.1.1.5 Send/Attach Feedback

A user should be able to send feedback about raw data he has worked with to it's originator in several ways:

- send an email to the originator of that entry (mailto: versus mail/submit web form)

- append a note/short message to the metadata record in the database which refers to the raw data. In this case it would also be necessary to have the list of notes appended to the metadata entry.

3.1.1.6 Wish list

Having a global wish list where database users/visitors can append comments/wishes on the system in general. The list should be stored in the database. The list must be viewable via the web. It needs to be checked whether messages need to be appendable specific to specific entries, aka threaded messages. The list shall only be deletable/editable by the system administrator. Entries shall be

reviewed by the administrator before they are put to the database and visible to the public. The administrator may become informed of new, pending entries. It needs to be checked whether this shall be implemented by a web form or just as a (viewable) mailing list.

3.1.2 Use cases by registered Web-Users

In addition to the above, registered users can perform several more operations. For registration the name and email address of the user is requested. Passwords should be sent by email, to reduce misuse.

3.1.2.1 Analyse Raw Data

In this scenario a privileged user (system administrator or owner/submitter of the data set) initiates the analysis of raw data whose metadata have been submitted recently. The user shall be able to select which analysis procedures will be applied by means of some check boxes. The list of possible analysis tasks is still open for discussion and depends on the format of the stored data. The implementation must be flexible enough to start with a small set of analyses, but can be extended also without detailed knowledge of the code. The system has to support also to cascade analyses, to include both, simple format conversions (e.g. binary/ASCII conversions) as well as the actual (e.g. statistical) analysis itself.

As this task takes some time to perform, it should

- run concurrently
- have a notification in the database saying it is active
- change that status when the analysis job is finished

Parts of this function may also be triggered automatically by the system.

3.1.2.2 Submit Data

This scenario allows the database user to submit a new metadata entry. This function must check that

- the user is allowed to post data
- all mandatory fields are filled
- all filled fields have meaningful values
- all URL fields have valid entriesTo be open for new data formats, it should be possible to add new description fields in addition to the defined common fields, as described in the table description (Section 3.2).

Further it should be possible to add also entries for ‘live’ measurements which produce results in future, and not only to already available measurement results. This feature has to be further investigated.

3.1.2.3 Modify Data

This scenario allows the database user to modify an existing metadata entry. This function must check that

- the user is modifying data he has previously submitted
- all mandatory fields are filled
- all filled fields have meaningful values
- all URL fields have valid entries

3.1.3 Use cases by System Administrators

System administrators are defined by the participants of the MOME project. Their main task is to keep the system working.

3.1.3.1 Manage Accounts

In order to manage the user accounts, following functions are needed:

- GUI for login/logoff
- Users can be added/deleted
- current login status needs to be checked prior to executing privileged functions

3.1.3.2 Manage Autonomous Processes

Some processes can be running as continuous services (see Section 3.1.4). System administrator(s) are responsible to manage these autonomous processes .

3.1.3.3 Modify Data

This scenario allows the system administrator to modify any existing metadata entry. This function must check that:

- all mandatory fields are filled
- all filled fields have meaningful values
- all URL fields have valid entries

3.1.4 Use cases by Computer (autonomous processes)

3.1.4.1 URL Verification

A process should take URLs stored in the database and check availability of these entities on the web. If not available a warning should be logged and possibly mailed to the system administrator. The error output should contain a link to the appropriate trace detail page. An indication of this error shall also be visible on the details page. The database entry has to be marked accordingly.

3.1.4.2 MD5 Auto Checker

If traces are downloaded for analysis an MD5 hash sum shall be computed and stored. This hash sum can be used for verifying the trace at a later stage. It helps to check, whether files located in the Internet have changed.

3.1.4.3 Page preparation, Page caching

A continuously running job may prepare static web pages (e.g. complete trace list info) so not each request by a user would trigger the complete database query and the complete process of building the html page from the result of the query. Techniques for such job should only be applied after it is clear that performance is a bottleneck in that sector.

3.1.4.4 Download Raw Data

After metadata entry for specific raw data has been entered, its URL needs to be validated and queued for download for later analysis. A cron-like job can then download the raw data for analysis at convenient times. Logging of the status for each database entry (unknown, checked, pending, downloaded, analysed) is needed. The status shall be visible within the detail page of the data set.

3.1.4.5 Preparation of Statistics

The analysis tools analyse the downloaded raw data and write their results into a section of the MOME database. Information about the tools to be applied depending on the raw data and metrics to be evaluated has to be extracted from the database . Information about successful execution of the analysis tasks shall be logged and made available to the administrator user. Results of analysis must be visible next to the detailed trace description. The values shall also be available as search criteria.

The system should be able to store user queries for later reference (i.e. Statistics on most popular queries, traces, etc.)

3.2 Table description

3.2.1 Initial table design

The use case study results in the need for independent tables to store the following information:

- Common attributes
- Additional attributes for the packet traces
- Results of analysis tasks applicable to the packet traces
- Additional attributes for the flow traces
- Results of analysis tasks applicable to the flow traces
- Additional attributes for QoS results
- Results of analysis tasks applicable to the QoS results
- Additional attributes for the routing data
- Results of analysis tasks applicable to the routing data
- Additional attributes for the HTTP traces

- Results of analysis tasks applicable to the HTTP traces
- Web-based data repository
- User Attributes

The only table used by all entries in the repository is the common attribute table. The other tables are used depending on the nature of the data associated to a given entry in the repository.

The following tables describe the different database tables of the MOME Database. The attributes for particular database tables are specified as follows. The “Attribute Name” denotes the name of the attribute in the database structure, while the “Data type” corresponds to its type. The mandatory attributes (marked with “Y” in the “Mandat.” column) must be filled by a user, who submits the meta-data entry. The “Overview” column in the common attributes table denotes, if particular attribute is presented at the overview screen to the users browsing the database. “Yes” in the “Filter” column denotes, that the users will be able to search the database using the value of corresponding attribute as the search criteria.

The 'analysis mode' field in the analysis tables is still to be defined. The possible variants are that the user manually introduces the results his analysis process or that a tool is used to automatically produce this data. When a tool is used, the process can be operator initiated or done automatically by the system. The definition of this mode heavily depends on the format of the data which have to be analysed. This will determine the availability of analysis tools.

Mand	***Attribute Name***	***Data type***	***Description***	***Filter***	***Overview***
Y	Data_set_name	Char	Identifier of the raw data set	Yes	Yes
Y	Data_type	Enum	packet trace, flow trace, QoS results, routing data, HTTP-trace, web-based data repository	Yes	Yes
	File_size	Int	File size in bytes	Yes	Yes
Y	Start_time	Datetime	Date and time of measurement start	Yes	Yes
Y	End_time	Datetime	Date and time of measurement end	Yes	Yes
Y	Submitter	Char	Id of the person who submitted the data	Yes	Yes
Y	Description	Text	Short description of the data set	Yes	Yes
Y	Data_location	Text	URL with the location of file with actual data	No	Yes
N	File_compression	Char	None, or: zip, gzip, tar,…	No	No
N	Md5_sum	Text	MD5 hash sum of raw data file	No	No
Y	Submission_date	Datetime	Date, when entry has been added	Yes	No
Y	Last_update	Datetime	Date, when entry has been updated or verified	Yes	Yes
N	Tool	Char	Name of the tool which generated the data	Yes	No

Table 3-1: Common attributes

Mand	*Attribute Name*	*Data type*	*Description*	*Filter*
Y	Network_type	Char	Trace collection environment, typical values: LAN, WAN	Yes
Y	Collector_location	Text	Location of the collector (country, city, institution)	Yes
Y	Traffic_type	Enum	Operational network traffic, artificial test traffic	Yes
N	Link_protocol	Char	Ethernet, VLAN, ATM, POS, WLAN,…	Yes
N	Link_speed	Int	In bits per second: 100M, 155M,…	Yes
N	Capture_mode	Char	Ethernet snooping, optical splitters, direct capture by router,…	Yes
N	Filter_rules	Char	None, or: only TCP packets, only packets with specific port number, only packets belonging to specific flow…	Yes
N	Number_packets	Int	Number of captured packets	Yes
N	Recorded_data	Char	IP header, TCP/IP header, entire packet, timestamp, packet size+timestamp,…	Yes
N	Trace_anonymisation	Char	None, or: IP address removed, IP address scrambled, IP address and TCP port scrambled, payload removed,…	Yes
N	Capture_platform	Char	linux, bsd, windows, router, DAG,…	Yes
Y	Data_format	Char	Libpcap, DAG, tcpdump, PSAMP, sFlow, fr, clr, tsh, other (URL to format description)	Yes
N	Additional_info	Text	Additional information on network and trace collection scenario	No

Table 3-2: Packet trace table

Analysis_task	*Data_type*	*Description*	*Analysis_ mode*
Avg_bit_rate	real[1]	Traffic rate, averaged over entire trace duration, in bit/s	TBD
Avg_pkt_int_time	real	Average packet inter-arrival time in sec	TBD
Avg_pkt_size	real	Average packet size in bytes	TBD
Avg_pkt_rate	real	Average packet arrival rate in pkt/sec	TBD
Pkt_size_dist	Graph/table[2]	Histogram of packet sizes	TBD
Protocol_dist	Graph/table	Bandwidth use per-protocol	TBD
Application_dist	Graph/table	Bandwidth use per-application	TBD
Rate_10ms	Graph/table	Series of rates (in bit/s), calculated over 10ms intervals	TBD
Rate_var_10ms	Real	Variance of bit rate calculated over 10ms intervals	TBD
Rate_1s	Graph/table	Series of rates (in bit/s), calculated over 1s intervals	TBD
Rate_var_1s	Real	Variance of bit rate calculated over 1s intervals	TBD
Rate_3min	Graph/table	Series of rates (in bit/s), calculated over 3min intervals	TBD
Rate_var_3min	Real	Variance of bit rate calculated over 3min intervals	TBD
Effective_bw	table	Effective bandwidth, calculated assuming different available bandwidth and buffer	TBD
Required_TB	table	Values of token bucket parameters, required for traffic in trace	TBD
Hurst_param	Real	Value of Hurst parameter, evaluating self-similarity	TBD

Table 3-3: Packet trace analysis result table

Mandat.	*Attribute Name*	*Data type*	*Description*	*Filter*
Y	Network_type	Char	Trace collection environment, typical values: LAN, WAN	Yes
Y	Collector_location	Text	Location of the collector (country, city, institution)	Yes
Y	Traffic_type	Enum	Operational network traffic(0), artificial test traffic(1)	Yes
N	Link_protocol	Cha	Ethernet, VLAN, ATM, POS, WLAN,…	Yes
N	Link_speed	Int	In bits per second: 100M, 155M,…	Yes
N	Capture_mode	Char	Capture by router (NetFlow), packet trace+analysis of flows,…	Yes
N	Filter_rules	Char	None, or only specific flows…	Yes
N	Number_flows	Int	Number of captured flows	Yes
N	Trace_anonymisation	Char	None, or: IP address removed, IP address scrambled, IP address and TCP port scrambled,…	Yes
N	Capture_platform	Char	linux, bsd, windows, DAG, router…	Yes
Y	Data_format	Char	IPFIX, NetFlow, IPDR, other (URL to format description)	Yes
N	Additional_info	Text	Additional information on network and trace collection scenario	No

Table 3-4: Flow trace table

1 will be substituted by the DBMS representation for double precision real numbers

2 depending on the experience gained from the prototype implementation, these objects might be stored directly on the database or, alternatively, on the server, in which case the database will contain a URL pointing to the file.

Analysis_task	*Data_type*	*Description*	*Analysis_mode*
Avg_flow_int_time	real	Average time between flow arrivals	TBD
Avg_flow_time	real	Average duration of a flow	TBD
Avg_flow_pkt	real	Average number of packets per flow	TBD
Avg_flow_bytes	real	Average number of bytes per flow	TBD
Avg_flow_arrival_rate	real	Average flow arrival rate in flows/s	TBD
Avg_traffic_rate	real	Average traffic rate in bit/s	TBD

Table 3-5: Flow trace analysis results

Mandat.	*Attribute Name*	*Data type*	*Description*	*Filter*
Y	Network_type	Char	Public Internet, private network, testbed network,…	Yes
Y	Measurement_type	Enum	Active,passive	Yes
Y	Metrics	Char	OWD, RTT, jitter, loss, throughput, etc…(multiple metrics possible)	Yes
Y	Sender_location	Char	Location of sender probe (or observation point in case of passive measurements)	Yes
Y	Receiver_location	Char	Location of receiver probe (or observation point in case of passive measurements)	Yes
N	Sender_platform	Char	linux, bsd, windows, router…	No
N	Receiver_platform	Char	linux, bsd, windows, router…	No
N	Timestamp_synch	Char	None, or: NTP, GPS,…	Yes
N	Number_values	Int	Number of recorded singleton measurements	Yes
N	Additional_info	Text	Additional information on measurement scenario	No
Y	Data_format	Char	Text or URL to format description	Yes

Table 3-6: QoS measurement table

Analysis_task	*Data_type*	*Description*	*Analysis_mode*
Avg_delay	real	Average delay (OWD or RTT, depending on which is measured)	TBD
Min_delay	real	Minimum measured delay	TBD
Max_delay	real	Maximum measured delay	TBD
10_delay_perc	real	10- percentile of delay	TBD
90_delay_perc	real	90- percentile of delay	TBD
Delay_dist	Graph (or table)	Histogram of measured delays	TBD
Avg_IPDV	real	Average delay variation	TBD
Min_IPDV	real	Minimum delay variation	TBD
Max_IPDV	real	Maximum delay variation	TBD
Loss_ratio	real	Packet loss ratio	TBD

Table 3-7:QoS measurement analysis result table

Mandat.	*Attribute Name*	*Data type*	*Description*	*Filter*
Y	Routing_protocol	Enum	BGP,RIP, OSPF, …	Yes
Y	Recorded_data	Enum	routing table, update packets	Yes
Y	Collector_location	Char	Location of the collector (country, city, institution)	Yes
Y	Data_format	Enum	Zebra, TCPDUMP [3]	

Table 3-8: Routing trace table

Analysis_task	*Data_type*	*Description*	*Analysis_mode*
Routing table size	Longint	The number of entries in the routing table	TBD
Routing updates	Longint	Number of update records in a file	TBD

Table 3-9: Routing trace analysis result table

Mandat.	*Attribute Name*	*Data type*	*Description*	*Filter*
Y	Collector_location	Text	Location of the collector (country, city, institution)	Yes
N	Filter_rules	Char	None, or only specific flows…	Yes
N	Number_entries	Int	Number of recorded entries	Yes
N	Trace_anonymisation	Char	None, or: IP address removed, IP address scrambled, IP address and TCP port scrambled,…	Yes
	Capture_platform	Char	linux, bsd, windows, DAG, router…	Yes
Y	Data_format	Char	Description, or URL to format description	Yes
	Additional_info	Text	Additional information on network and trace collection scenario	No

Table 3-10: HTTP trace table

3 At the point of writing this document, there are only two possibilities to create traces with routing data:
* The Zebra Open Source Router is able to dump BGP-4 routing tables and routing traffic to files. This format is shared with other projects (i.e. MTRD) and documented in the Zebra Project Documentation.
* TCPDUMP can be programmed to filter and store routing traffic. The filters can separate routing packets depending on the routing protocol.

Analysis_task	*Data_type*	*Description*	*Analysis_mode*
Avg_HTTP_req_int_time	real	Average time between flow arrivals	TBD
Avg_HTTP_response_pkt	real	Average response size in packets	TBD
Avg_HTTP_resp_bytes	real	Average response size in bytes	TBD
Avg_HTTP_request_rate	real	Average request arrival rate in flows/s	TBD

Table 3-11: HTTP trace analysis result table

Man dat.	*Attribute Name*	*Data type*	*Description*	*Filter*
	baseurl	Text	URL of the main page of the repository (if differs from the common URL)	Yes
Y	data_provider	Text	Name of the providing institution	Yes
Y	measurements	Text	Available measurements (delay values, rtts, routes, packet loss, link utilization, …)	Yes
Y	raw_data	Enum	Raw data available (yes, no, partially)	No
	refresh_period	Enum	Daily/Weekly/Monthly/Yearly/Irregularly	Yes

Table 3-12: Web based data repository table

Mand.	Attr. name	Data type	Description
Y	ID	Int	unique
Y	username	text	some arbitrary username
Y	name	text	Full name of the user
Y	date	datetime	when was entry made
N	status	enum	Enabled/disabled
Y	accessLevel	enum	admin or normal
Y	email	text	email address
N	homepage	text	homepage address
N	description	text	optional user description
Y	lastLogin	datetime	set automatically
Y	logins	Int	number of logins
Y	password	blob	Not in plain text

Table 3-13: MOME user table

The MOME user table is shared with the MOME tools database designed in WP1 and will be described in Deliverable D12 “MOME Interoperability Database”.

3.2.2 MOME database extensibility

The MOME database is extensible by inserting more fields into the previously described tables and defining and including new tables describing new data models or analyses.

3.3 The MOME database data model

Figure 3-1 shows the data model of the MOME database derived from the table definitions and the use cases.

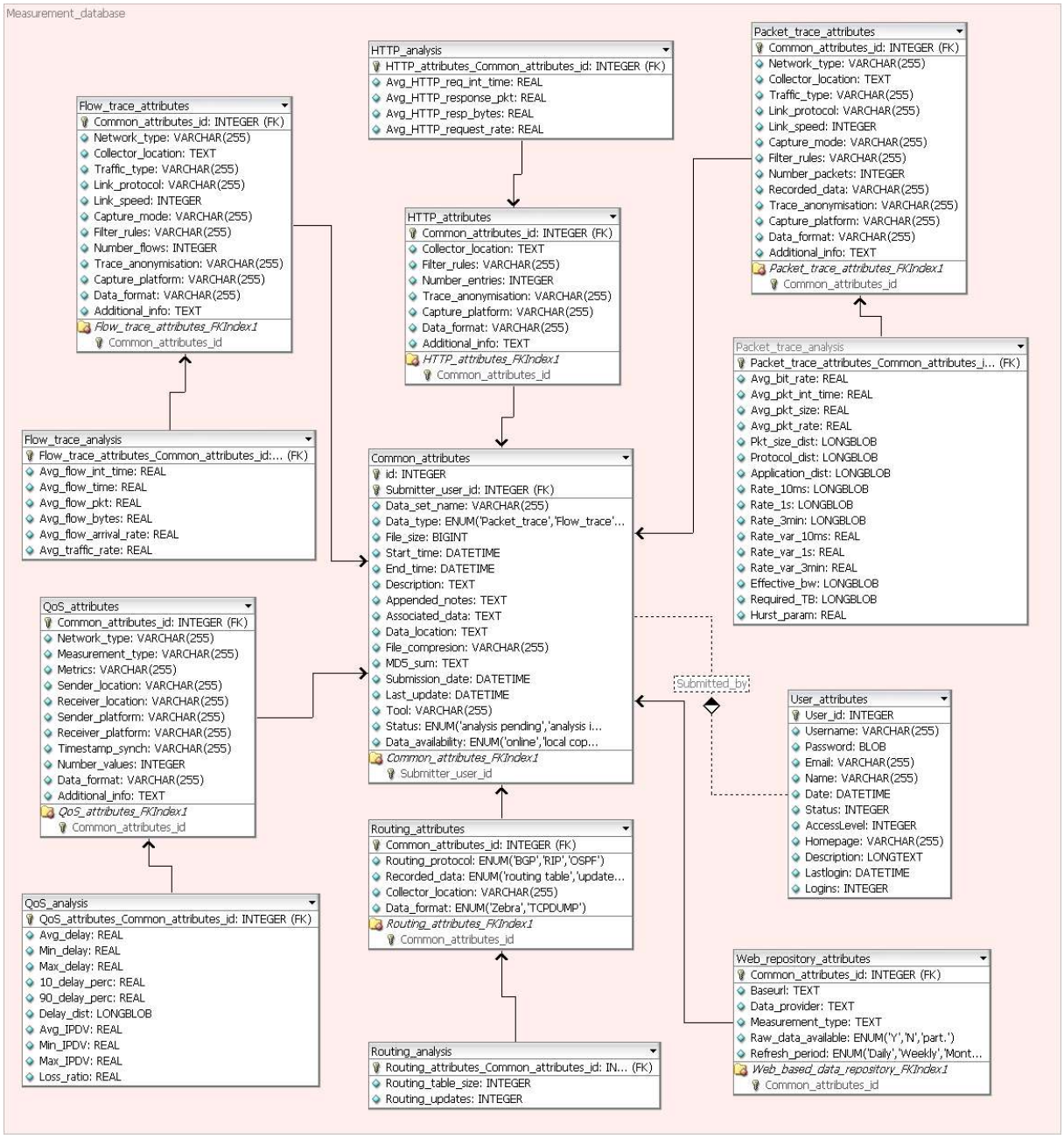

Figure 3-1: The MOME database data model

The entities in the database are each described by his own table. The relationships between entities are as follows:

- a user can submit one or more entries (described by an element of the common_attributes table)

- an attribute depends on the existence of its entry (in the common_attributes table)

- an analysis depends on the existence of its attribute

4 DBMS requirement analysis

4.1 DBMS user interface

The MOME Database will use a Web based user interface to allow easy user access. All DBMS operations will be hidden to the user.

4.2 DBMS interface to the WWW server.

PHP scripting managed by the WWW server will be used to provide a flexible programming environment for the WWW MOME database GUI. PHP scripting integrates well both with commercial, closed source and free, open source DBMS implementations.

4.3 DBMS selection

Due to the nature of the activities related with the MOME database , there is no real need for a commercial DBMS. Open source DBMS implementations like PostGreSQL and MySQL don't lack behind closed source DBMS for the amount of data to be handled in the initial project phase.

Support for SQL is the only important factor to ensure portability. Minor SQL idiom variants between the different vendors will have to be taken into account if a DBMS migration is needed, but a minimum layer of interoperability is ensured. Furthermore, most DBMS design tools are able to generate the DBMS specific SQL scripts from a common design.

4.4 Hardware requirements

The main challenge for the MOME Measurement Database is the amount of storage needed for local copies of measurement datasets, since the initial design of the database requires that the datasets be stored in it. This section studies storage requirements for the operating system and some types of data hosted in the MOME database.

4.4.1 Storage requirements for the operating system and applications

The software specifications for the MOME databse Web server include

- Operating system: Debian 3.0 Linux
- Web server software: Apache 1.3.26 (with mod_php4, apache_ssl, etc.)
- DBMS MySQL 3.23.49 (phpMyAdmin 2.2.3 as management interface)

These result in a storage requirement of 4 Gbytes on the Web Server platform and Operating system.

4.4.2 Storage requirements for the MOME database

The MOME database supports two kinds of data: metadata, i.e. attributes of measurement data and analysed data, which can be connected to each of the meta-datasets. The storage requirements for metadata depends on the length of the entries. Supposing that all VARCHAR type variables take 256 bytes of storage space, and all TEXT and BLOB type variables take 32768 bytes, we get the following storage requirements:

	Packet traces	***Flow traces***	***QoS results***	***Web repositories***	***Routing***	***HTTP traces***	***Mean size***	***Entries per GB***
Base size	207000	207000	174000	174000	139000	205000	194900	5500
Size w/ analysis	471000	209000	209000	174000	141000	207000	245800	4400

Expecting that some datasets will have an analysis included and some will not, the first GB of data including database overhead is reached with approximately 5000 metadata entries.

4.4.3 Conclusions

To allow a comfortable operation of the MOME database during the MOME project duration and beyond, a state of the art disk with a capacity of around 80 Gbytes is selected.

Storage requirements for the operating system and the applications amount to around 4 Gbytes of data, which is 5% of the complete disk storage space. 1 Gbyte of metadata allows the storage of around 5000 entries in the meta-database. Therefore a disk with 80 Gbytes allows a reasonable allocation of disk space for locally hosted traces.

Finally, the Dell Poweredge 1750 server blade, which is standardised in the TERENA data centre, was chosen. It's main features are:

- 1 Intel Xeon 3.0 GHZ CPU (1 Mb cache)
- 1024 Mb DDR SDRAM
- 146 Gb U320 10K SCSI harddisk

RAID is not configured adapter since the system recovery is granted by full daily backups and hot swappable disks. Future growth of the MOME database is also accounted for. The selected system has two empty slots for further extensions.

5 Database prototype implementation

To enable rapid prototyping a light-weight PC is installed at Salzburg Research, which provides a similar environment as the final implementation, but lacks large amount of disk space, computation power and high performance network connectivity.

Prototype Main Parameters:

- Pentium PII 350MHz, 128MB RAM, 6 GB HD

- Linux Kernel 2.4.18
- Apache 1.3.26 (with mod_php4, apache_ssl, etc.)
- MySQL 3.23.49
- phpMyAdmin 2.2.3

The following figures show screenshots from the phpMyAdmin GUI.

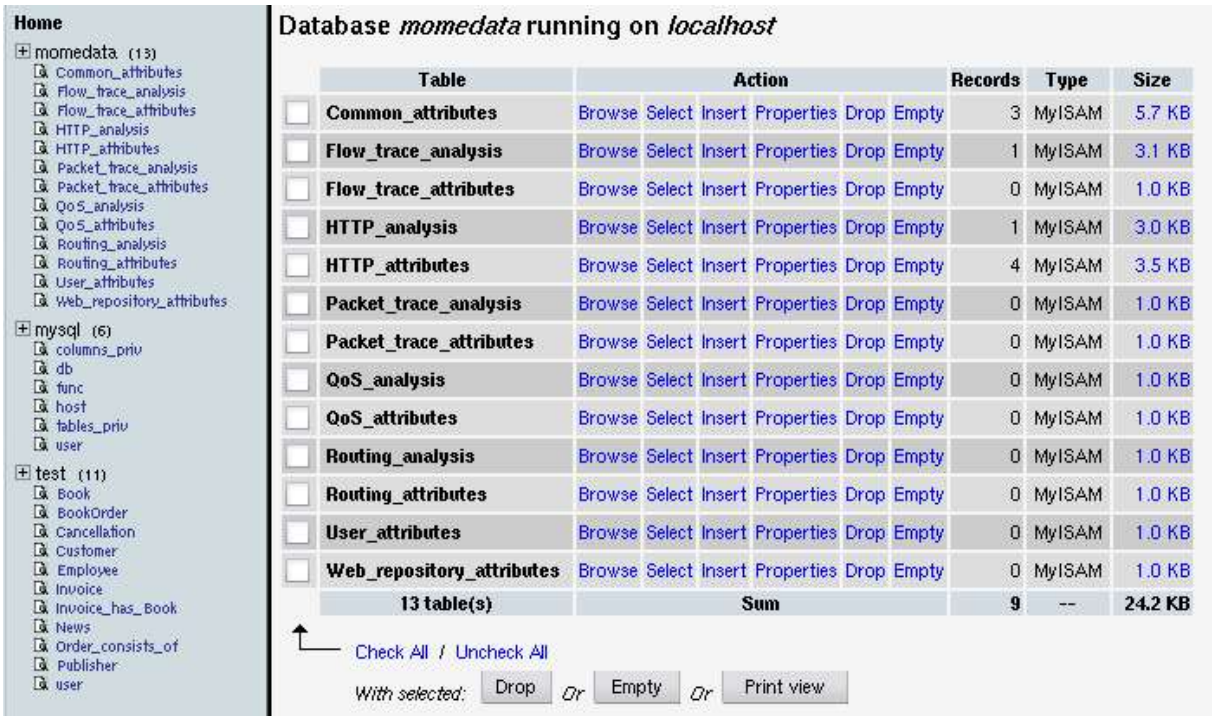

Figure 5-1: Table view

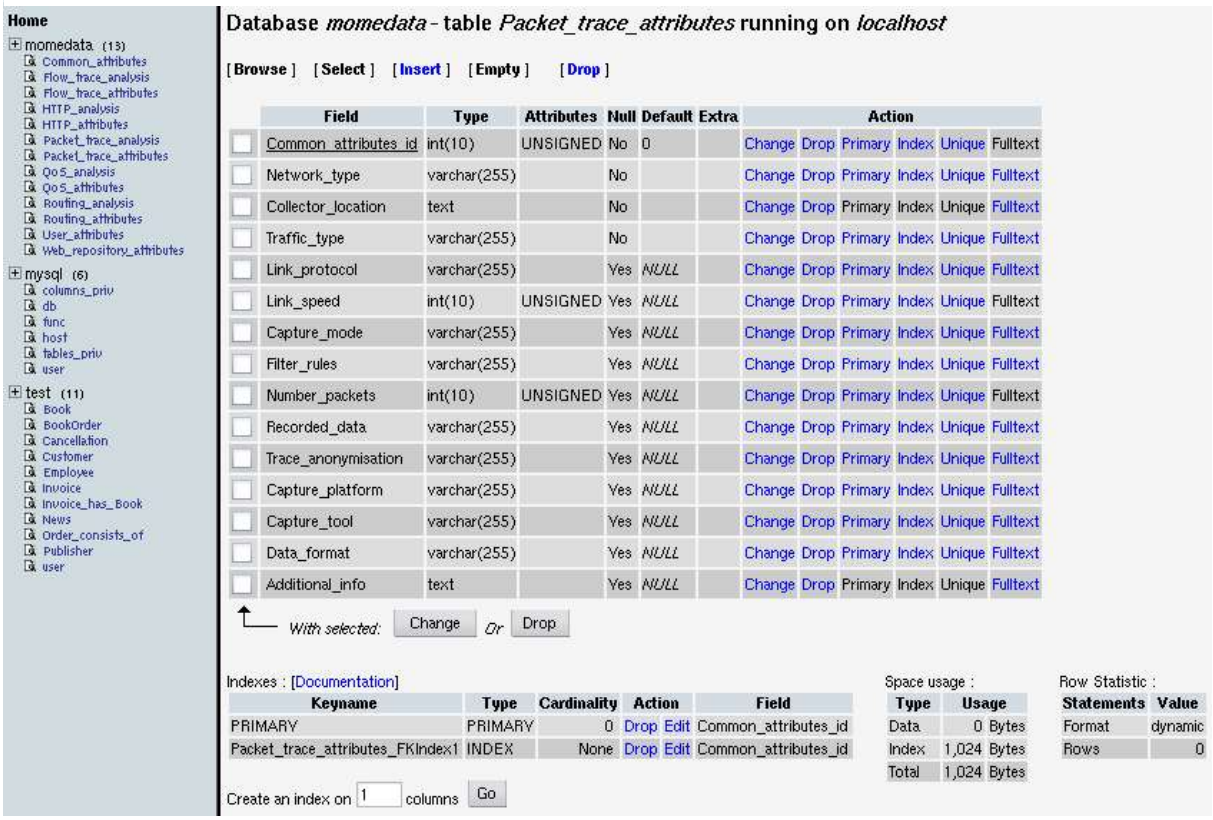

Figure 5-2: Packet trace table view

Annex A Survey of existing data repositories

A.1 Internet Traffic Archive

Category: traffic traces repository.

The archive is accessible by: http://ita.ee.lbl.gov/index.html

Description

The Internet Traffic Archive is a repository of traces, sited at the Lawrence Berkeley National Laboratory. All of the traces are freely accessible. However, some of them have restrictions on redistribution. The traces were produced by different institutions and on different sites, during time period from 1989 to 1998.

Available data sets

The following packet, TCP and HTTP trace sets are available:

- BC: 4 million-packet traces of LAN and WAN traffic seen on an Ethernet.
- DEC-PKT: 4 hour-long traces of all wide-area packets.
- LBL-TCP-3: 2 hours of wide-area TCP packets.
- LBL-PKT: 2 hour-long traces of all wide-area packets.
- LBL-CONN-7: 30 days of wide-area TCP connections.

- WorldCup98: 1.3 billion Web requests recorded at servers for the 1998 World Cup.
- EPA-HTTP: a day of HTTP logs from a busy WWW server.
- SDSC-HTTP: a day of HTTP logs from a busy WWW server.
- Calgary-HTTP: a year of HTTP logs from a CS departmental WWW server.
- ClarkNet-HTTP: two weeks of HTTP logs from a WWW server.
- NASA-HTTP - two months of HTTP logs from a busy WWW server.
- Saskatchewan-HTTP - seven months of HTTP logs from a University WWW server.
- BU-Web-Client - Six months of Web client traces.
- UC Berkeley Home IP Web Traces - 18 days of HTTP traces.
- NPD-Routes - Two datasets of repeated Internet route measurements.

Most of the traces are available as ASCII files, compressed and stored on FTP server.

metadata and statistical info

Each trace set is described by the following metadata:

- Description: short description of the network scenario and type of collected data
- Format: detailed format of the trace file
- Measurement: description of the measurement scenario
- Privacy: how the privacy of information is ensured
- Acknowledgements: to people who contributed to collecting the traces
- Publications: references to papers on research done using the traces
- Restrictions: comments on possibility to re-distribute the traces
- Distribution: links to the actual data

No additional statistical information about the captured traffic is available.

A.2 MAWI Working Group Traffic Archive

Category: traffic traces repository.

The archive is accessible by: http://mawi.wide.ad.jp/mawi/

Description

The traffic data repository is maintained by the MAWI Working Group of the Japanese project WIDE. It contains packet traces from the WIDE backbone. Traffic traces are captured using tcpdump. Traces are freely accessible only for research purposes.

Available data sets

Available traffic traces were captured at the following locations:

- sampling point-A: a trans-Pacific T1 line
- sampling point-B: another trans-Pacific line
- sampling point-C: an IPv6 line connected to 6Bone
- sampling point-D: an IPv6 line connected to WIDE-6Bone
- sampling point-E: another US-Japan line (OC-3)

The detailed trace capture scenario is not known. Each dataset contains a number of daily traffic traces, taken on different days, starting from 1999 and continuing up till now (2004). Traces are stored in tcpdump format, as compressed files on FTP server.

Metadata and statistical information

Each trace is described by the following metadata:

- File information: file name, file size, id
- Trace information: start time, end time, total time

Additional statistical data, derived from the trace, are:

- Average rate
- Number of flows
- Average flow size
- 10 biggest flow sizes
- Number of IP addresses
- Packet size distribution (plot and table)
- Protocol breakdown (plot and table)

A.3 NLANR Network Traffic Packet Header Traces

Category: traffic traces repository.

The archive is accessible by: http://pma.nlanr.net/Traces/

Description

NLANR PMA centre is located at the University of California, San Diego. The PMA data collection infrastructure consists of a number of monitors, located at aggregation points within large US networks: vBNS, Internet2 / Abilene. Most of the monitors are located in POPs, or on the links between the University campuses and the wide area networks. The detailed descriptions of the monitor locations are provided on the accompanying web-page.

Available data sets

The repository contains daily updated traces from 17 data collection sites. The measurement strategy is to capture samples eight times a day, for a defined length of time (currently 90 seconds). Traces are kept as compressed files in a directory structure, which can be browsed by web. Traces from each day are kept in a separate directory. Currently, traces from the last month are available.

Traces contain the time-sequenced records of IP headers and are stored in compressed binary formats (fr, fr+, crl, tsh). Additional tools are available for converting the binary files to ASCII. Files are accessible by HTTP.

Metadata and statistical information

The date of the trace collection is coded in a name of the directory. The file name contains the symbol of a trace collector location. Additional information about the network and the collection scenario is kept on a separate web-page for each location. Concluding, most important information is available, but finding it is quite troublesome and requires browsing through the file directory and different web-pages.

It is possible to search the trace catalogue, by the following criteria: trace duration, number of hosts, number of packets, megabytes of data

Additionally, statistical information for some of the traces is available, although it must be accessed and viewed using separate web-page. The database of statistical data can be searched by the location of the collector, collection date, and type of available statistical information:

- Top throughput
- Flow data
- Packet length statistics
- Flow summary
- Transaction summary
- Rate
- TCP flag analysis
- Packet length run length classes
- Asymmetric analysis

A.4 Waikato Internet Traffic Storage

Category: traffic traces repository.

The archive is accessible by: http://wand.cs.waikato.ac.nz/wand/wits/

Description

The repository is maintained by the WNAD research group from the University of Waikato, New Zealand. The traces were collected at the campus network of University of Auckland and at the New Zealand Internet Exchange.

Available data sets

The following trace collections are available:

- Auckland-I: 1 week of IP headers collected on July 1999 at the University of Auckland uplink
- Auckland-II: 42 1-day traces collected from Dec 1999-Jun 2000 at the University of Auckland
- Auckland-IV: a 6 1/2 week trace from Feb and Apr 2001 at the University of Auckland uplink
- Auckland-VI: 4 1/2 days three point trace taken in June 2001 at the University of Auckland
- A Local ISP: a collection of traces taken at a local ISP between Nov 1999 and Jan 2000
- NLANR: copies of traces taken by the NLANR MOAT team during Nov 1998 and Feb 2000
- NZIX-I: 228 10 minute IP header traces taken at the New Zealand Internet exchange between December 1998 and April 1999
- NZIX-II: 16 traces taken at the New Zealand Internet exchange in June/July 2000

Most traces consist of the time-stamped IP headers and are either in the DAG format, or in the format of proprietary trace collection software. Format conversion tools are available. Most of the traces are available on request, on magnetic tapes. Some sample traces are also publicly accessible via FTP.

Metadata and statistical information

Network scenario and trace collection scenario are described in detail. Each trace file is described by the run length, number of collected IP headers and file size. For most of the traces some derived statistical data is available, in the form of graphs presenting time-plots of:

- Rate in packet/s (calculated in 1 sec intervals), separately for each protocol (UDP, TCP) and application (HTTP, telnet…)
- Rate in bit/s (calculated in 1 sec intervals), separately for each protocol (UDP, TCP) and application (HTTP, telnet…)
- Rate of new connection arrivals (in connections/s)
- Number of active connections
- Average connection duration and volume (in packets and bytes)

A.5 NLANR Measurement and Network Analysis

Category: end-to-end QoS measurement data

The archive is accessible by: http://watt.nlanr.net/active/maps/ampmap_active.php

Description

The measurements are performed by the Active Measurement Project (AMP) architecture, which consists of approximately 150 active monitors, deployed on high-speed research networks throughout the United States, as well as at strategic sites in other countries. Across all sites, round trip time (RTT), packet loss, topology, and throughput (user/event driven) are currently measured, using the IPMP protocol.

Available data sets

The collected data is available by a web interface, where users can select the source and destination probes. The raw data has the format of two columns. The first column denotes the seconds elapsed since the beginning of the start of the day, and the second column is the RTT or the word “loss” if the probe packet was lost.

The user can view the weekly and daily (including the current day) graphs of RTT, loss, IP hop count, jitter, and distribution of RTT. There are daily graphs for every day since measurement began. The graph for the current day is live.

Metadata and statistical information

The detailed information on the source and destination host is available. Additional available information includes the trace of the route, recorded by the trace-route utility at the time of the actual measurement. Statistical information derived from the data can be viewed also in the form of time plots and it includes:

- Distribution of RTTs
- Weakly averages
- Weakly 10- and 90- percentiles

A.6 Abilene NetFlow Nightly Reports

Category: flow data

The archive is accessible by: http://www.itec.oar.net/abilene-netflow/

Description

Daily summaries of NetFlow reports from the Internet2 Abilene backbone network.

Available data sets

The raw NetFlow data sets are available only on request. Public access is offered to daily and weekly summaries. Reports from all 15 routers of Abilene backbone are available, for every day since

September 2003. Reports are created by the “flow-tools” software and can be viewed in the form of HTML table or plain ASCII format.

Metadata and statistical information

There is not much additional information regarding the scenario of collecting the statistics, besides the date and the name of the city, where the router is located. Large amount of statistical information is available, which includes:

- Observation time
- Average flow time
- Average rate in pkts/sec
- Average flow rate in flows/sec
- Average number of pkts/flow
- Distribution of IP packet size
- Distribution of number of packets/octets per flow
- Distribution of flow time

The data used for calculating the above statistics can be filtered by: routing information (AS number, source-, dst- prefix), transport protocol (TCP and UDP, port numbers), router information (input interface, output interface), TOS value, multicast information (source group, input interface)

A.7 Abilene Routing Data

Category: routing data repository

The archive is accessible by: http://abilene.internet2.edu/observatory/data-collections.html#routing

Description

The repository of BGP data from the Internet2 Abilene backbone network. Abilene has deployed the Ixia IxTraffic System that BGP peers with all of the Abilene backbone routers. It records all of the BGP routing information it learns from the routers.

Available data sets

The data is stored in two files for each day.

- The "ribs" file is a snapshot of the entire BGP RIB taken once a day.
- The "events" file is a list of all the changes that happened during the day.

With both files, one can form a complete picture of what happened to BGP routing on the network over the course of the day. The snapshots were taken on each day from 5.12.2002 till 1.1.2004. All files are stored in a common directory and are accessible by HTTP in ASCII format.

Metadata and statistical information

Besides the date of capture, there is no additional information on the stored data. There is also no additional statistical data.

A.8 RIPE NCC Routing Information Service Raw Data

Category: routing data repository

The archive is accessible by: http://data.ris.ripe.net/

Description

This repository contains raw data collected by the RIPE routing collectors. The data is collected using Zebra, and it is stored in MRT format.

Available data sets

The data from the following collector locations is available:

- RIPE NCC, Amsterdam. Collects default free routing updates from peers. From October 1999.
- LINX, London. Collects route updates announced by LINX members. From July 2000.
- SFINX, Paris. Collects route updates announced by SFINX members. From March 2001.
- AMS-IX, Amsterdam. Route updates announced by AMS-IX members. From January 2001.
- CIXP, Geneva. Collects route updates announced by CIXP members. From April 2001.
- VIX, Vienna. Collects route updates announced by VIX members. From June 2001.
- Otemachi, Japan. Collects route updates announced by JPIX members. From August 2001.
- Stockholm, Sweden. Route updates announced by the NETNOD members. From April 2002.
- San Jose, USA. Route updates announced by the MAE-WEST members. From May 2002.
- Zurich, Switzerland. Collects route updates announced by the TIX members. From May 2003.
- Milan, Italy. Collects route updates announced by the MIX members. From Nov 2003.
- New York (NY), USA. Route updates announced by the NYIIX members. From Feb 2004.

Two sets of files are available for each of the collectors:

- Trace of all BGP packets, created with the Zebra command "dump bgp all ...". The files are created every 15 minutes.
- The entire BGP routing table, created with the Zebra command "dump bgp routes-mrt ...". These files are created every 8 hours.

The compressed files are stored in a directory structure, where folder name corresponds to the data of the snapshot. Files are accessible by HTTP.

Metadata and statistical information

The collection scenario describes some settings of the data collection software. The date of the snapshot and the location of the collector are also specified. There is no additional statistical data.

A.9 University of Twente M2C Measurement Data Repository

Category: traffic repository

The archive is accessible by: http://m2c-a.cs.utwente.nl/repository/

Description

This repository stores anonymised packet traces taken from various locations in the Netherlands. The traces are stored in libpcap/tcpdump compatible format

Available data sets

The data from the following collector locations is available:

- Measurements taken in July 2002 on the 300 Mbit/s link between the residential network of a University in the Netherlands and the core network of the University of Twente.
- Measurements taken between May and August 2003 on the 1Gbit/s link of a dutch research institute to the Dutch academic and research network.
- Measurements taken from September to December 2003 on the 1 Gbit/s access link to the Dutch academic and research network of a large college
- Measurements taken between February and July 2004 on the aggregated uplink of an ADSL access network.

A.10 EuroNGI (Design and Engineering of the Next Generation Internet -Towards convergent multi-service networks)

Description

The EuroNGI project [20] is a Network of Excellence within the 6FP IST program. The investigated research topics include, among others, IP traffic measurements, characterization and data analysis.

EuroNGI plans to establish a common repository of traffic traces. This repository will enable sharing of available traces by project partners and common research on traffic characterization and modeling. The activities of EuroNGI project can benefit from the MOME database, which will provide access to other measurement data repositories, described in detail by additional meta-data.

The project's web page is http://eurongi.enst.fr/en_accueil.html

A.11 List of other identified data repositories

Traffic traces:

- http://www.comet.columbia.edu/~veres/DATASETS/Datasets.html: collection of TCP traces
- http://www1.cs.columbia.edu/~wenyu/papers/vad_samples/list.html: Packet Radio traces
- http://www-tkn.ee.tu-berlin.de/research/trace/trace.html: MPEG-4 and H.263 video traces
- http://trace.eas.asu.edu/: MPEG-4 video traces
- http://www.web-caching.com/traces-logs.html: WEB traces and logs
- http://research.cs.vt.edu/nrg/traces.html: trace files of WWW traffic
- http://netserv.iet.unipi.it/TeleTraffic/: TLC Netgroup LAN traffic traces
- http://www.csee.usf.edu/~christen/career/software.html: traffic modeling project
- http://www.caida.org/analysis/measurement/oc48_data_request.xml: CAIDA OC48 data (not publicly accessible)
- http://www.ncstate.net/nts/research/traffic/: NC State University and Duke University traces
- QoS measurements (most of the sites offer only statistical data):
- http://ipmon.sprint.com/delaystat/: SPRINT IPMON delay analysis
- http://www.internettrafficreport.com/main.htm: Internet traffic report
- http://ipnetwork.bgtmo.ip.att.net/pws/index.html: AT&T network performance
- http://www.internetpulse.net/: Internet health report
- http://global.mci.com/about/network/latency/: MCI latency statistics
- http://abilene.internet2.edu/observatory/data-collections.html: Abilene observatory
- http://www.caida.org/tools/measurement/skitter/research.xml: CAIDA skitter data
- http://www.ripe.net/ttm/Plots/: RIPE NCC test traffic measurements (not publicly accessible)
- http://measurement.internet2.edu/traceroute.shtml: Internet2 traceroute and ping

Routing data:

- http://archive.routeviews.org/: University of Oregon Route Views Archive project
- http://moat.nlanr.net/AS/Data/: NLANR Archived Route Views Data
- http://www.pch.net/documents/data/routing-tables/: Packet Clearing House Archived Route Views Data

- http://www.caida.org/tools/measurement/skitter/router_topology/: CAIDA topology measurements
- http://noc.ilan.net.il/LG/: ILAN looking glass
- http://bgp.lcs.mit.edu/: MIT BGP monitor

Annex B State of the art in data analysis

This annex presents the review of known measurement data analysis goals and methods. The aim of the review was to identify the analysis tasks, possible to integrate in MOME database.

B.1 Pre-processing of data

Data produced by the pre-processing tools usually do not present valuable information by itself. However, they can be used as the input data for other analysis tools. Some of the methods for performing the pre-processing are listed below:

- **Format conversion**, which do not change the content of information, but just convert one storage format to another one. Exemplary format conversions are: from a proprietary format to some standard format, or from a binary data to ASCII file.
- **Filtering**, which allows us for extracting a needed data from the entire data set captured by measurements. In this case, the input consists of the entire data set (e.g. a traffic trace or some series of QoS result values), while the output is the sub-set of the original data. Possible filtering criteria are:
    - For filtering packets from the traffic trace:
        - By source and destination address/prefix
        - By protocol (TCP, UDP, RTP…)
        - By source and destination port number
        - By TOS/DSCP value
        - By packet size
        - By time (e.g. all packets observed within specified period)
    - For filtering flows from the traffic trace:
        - By size (number of packets or duration)
        - By rate (in packets/s or bits/s)
        - By time (e.g. all flows observed within specified period)
    - For filtering results from the QoS results set:
        - By threshold (e.g. all OWD values greater than given threshold)

  - By time (e.g. all results obtained within specified period)

- **Joining**, which is an opposite operation to filtering. The pre-processing tool takes as input data a couple of trace files (e.g. captured at the same location but for two separate transmission directions), and joins captured records to produce a single trace.

- **Calculation of rates** within specified intervals. Usually, the original traffic trace is represented as a vector of time-stamped records of packets, captured at the observation point. However, sometimes it is needed to convert this trace into the vector of rates (in bits/s), calculated over consecutive, non-overlapping intervals of certain length.

- **Sampling** allows for reducing the amount of result data, which has to be stored and possibly transmitted in the network.

- **Privacy protection**, which includes anonimization of the traffic trace, e.g. by removing or scrambling the IP address information in packet headers.

- **Routing data analysis** includes processing the routing tables and update messages, to obtain the information about the logical topology of the network. The topology may me needed e.g. as an input information for traffic matrix estimation tools.

Exemplary tools (more details can be found in [4]):

- http://www.packetfactory.net/Projects/ngrep/, packet capturing tool with ‘grep’ filtering.

- http://masaka.cs.ohiou.edu/~eblanton/tcpurify/, ‘tcpurify’.

- http://m2c-a.cs.utwente.nl/tools/throughput-alltimescales.pl , calculation of rates in arbitrary time-scales.

- http://www.tcpdump.org/other/tcpslice.tar.Z , creating subsets and supersets of traces.

B.2 Traffic analysis and modelling

This chapter describes the methods of processing the measurement data for obtaining certain statistical information. The purpose of the analysis is to understand the characteristic features of the data and to obtain aggregated information on the traffic, in the form of a set of model parameters.

Traffic in packet networks can be modelled on different levels, e.g. at packet level, connection level or session level. On each level, one can analyse and model the arrival process (of packets, connections, sessions) and duration characteristics (packet lengths, connection and session durations). Therefore, a proper identification of particular flows and sessions becomes an important issue. Usually it is assumed that the notion of a connection can be related to a single TCP connection or sequence of UDP packets belonging to a particular stream. From a traffic trace, connections can be recognized by inspecting the IP header fields. The notion of a session is more related with the user behaviour, and thus the sessions are more difficult to distinguish in the measured packet trace.

B.2.1 Calculating empirical statistical parameters and distributions

This task consists of calculating the empirical statistical parameters of the data set, without assuming any particular model or analytical distribution.

The input data is treated as a statistical sample. The data can be of different type, depending on the type of measurement and the considered level of traffic model. For example, when the measured data is a packet-level trace, the analysed data can consist of a set of observed:

- Packet lengths
- Packet inter-arrival times
- Rates, calculated over specified intervals

For flow-level traces, the analised data can consist of a set of:

- Flow durations
- Flow inter-arrival times
- Number of packets per flow
- Number of bytes per flow

For session-level traces, the analised data can consist of a set of:

- Session durations
- Session inter-arrival times
- Number of flows per session

For the QoS measurement results, the analised data can consist of a set of:

- Measured singleton OWD values
- Measured singleton RTT values
- Measured singleton IPDV values

Notice, that some of the above input data sets (e.g. flow inter-arrival times), can be obtained by properly pre-processing the original packet trace (filtering-out packets belonging to specific flows and recognizing flow start times). Of course, this can be done only if enough information is available in the original trace. In particular, the packet headers have to be kept. Another way to produce the required data is to use the measurement tool, which automatically recognizes and records the flows, e.g. Netflow.

As the output data, the following parameters can be calculated:

- Sample size (number of observed packets, trace duration, number of flows…)
- Average value
- Sample variance
- Other moments
- Empirical autocorrelation (if the data set is a time-series)

- Quantiles (for obtaining the histogram)

The analysis of properly filtered traces allow us to get additional information about the network use:

- Per-protocol bandwidth use (total number of bytes, or average rate measured for different protocols: TCP, UDP, ICMP…)

- Per-application bandwidth use (total number of bytes, or rate measured for different applications: WWW, FTP, telnet, VoIP, p2p…)

- Per-subnet bandwidth use (total number of bytes, or rates measured for traffic with different IP address prefixes)

Additionally, some protocol-specific statistics can be calculated, e.g. in the case of TCP:

- Statistics of TCP Options

- Average TCP congestion window

- Length of bursts of duplicated data

- Length of bursts of Out-of-Sequence data

- Average RTT

Exemplary statistical tools (more details can be found in [4]):

- http://www.tlc-networks.polito.it/diana/, framework for statistical analysis of traces.

- ftp://tracer.csl.sony.co.jp/pub/mawi/tools/tcpd-tools.tar.gz, statistical analysis of traces.

- http://www.tcptrace.org, statistical analysis of traces.

- http://members.aol.com/johnp71/javasta2.html, collection of statistical analysis tools.

- http://www.statsci.org/free.html, collection of general-purpose statistical analysis tools.

- http://www.r-project.org/, general-purpose statistical package.

- http://cm.bell-labs.com/cm/ms/departments/sia/software/index.html, statistical tools.

- http://www.isi.edu/saman/ramp.html, calculating empirical CDFs of various parameters of traffic model

B.2.2 Fitting to probability distributions

Fitting the empirical data to the known probabilistic distributions enables developing thorough analytical traffic models. The analysis should attempt to answer the following question: which distribution, and with what parameter values, best describes the studied data set. The well-known methodological approaches to this task are:

- Probability plotting. The methods for drawing quantile plots are known for a number of distributions: Normal, Log-Normal, Gamma, Chi-square, Exponential, Weibull.

- Tests for distributional assumptions. The known tests include the W tests (for Normal and Log-Normal distributions) and Chi-squared goodness-of-fit-tests (for any distribution).

The input data for the analysis can be different, similarly as in the case of empirical statistical parameters. However, the output data is now the answer, whether the particular specification effectively describes the empirical measurement data. Some typical distributions often assumed in traffic studies are for example: exponential distribution for modelling inter-arrival times of flows or sessions, exponential or heavy-tailed (e.g. Pareto) flow durations, Weibull distribution of packet delays in a router.

B.2.3 Fitting parameters of traffic models

Some special traffic models were developed in the literature for description of the traffic arrival process in packet networks. These models try to capture the multi time-scale variability and correlation structure of traffic. The appropriate analytical model, validated and parametrised with the help of real measurement data, can be used as a prediction of traffic for the purpose of resource dimensioning, or for generating artificial traffic according to realistic model in the simulation experiments. Some exemplary models, together with references to the literature on the methods of their parameter estimation from the trace, are listed below:

- Fractional Brownian Motion (FBM) models [5]

- Multi-fractal models [5]

- Multi time-scale models based on Markov-Modulated Poisson Processes (MMPPs) [9]

- Models based on Batch Poisson Arrival Processes (BMAPs) [9]

In this case, the input data is either a time-stamped trace of packet/flow arrivals, or a vector of rates calculated over specific intervals. The output data consists of the calculated parameters of the assumed model.

Exemplary tools (more details can be found in [4]):

- http://www.ip2bmap.de/, fitting parameters of BMAP-based traffic model.

- http://spin.rice.edu/DARPA/soft.html#MWM, fitting parameters of multi-fractal models

B.2.4 Validation of self-similarity

It is commonly believed, that the Internet traffic has so called self-similarity property. It means, that it is highly variable on multiple time scales. A number of methods are known, which attempt to answer, if the studied traffic is self-similar, or not. The known methods for assessing the self-similarity and estimating the value of Hurst parameter include ([5], [7]):

- Analyzing the plot of Index of Dispersion for Counts (IDC)

- Analyzing the variance-time plot

- The R/S analysis: the pox diagram method

- Periodogram-based analysis of the measured data in frequency domain

- AV estimator: wavelet-based analysis

These methods take as input the series of packet/flow arrival times, or series of rates, and produce as output the value of Hurst parameter, or the plot, from which this value can be estimated.

Exemplary tools: http://www.cubinlab.ee.mu.oz.au/~darryl/secondorder_code.html, the MATLAB codes for the wavelet-based estimation of self-similarity

B.2.5 Traffic prediction

Traffic prediction aims to foresee future behaviour of traffic, based on observation of the past and using appropriate model. Notice, that the methods of prediction can be applied for the time series representing the traffic load (e.g. prediction of traffic for the purpose of traffic engineering), or to the series of results of QoS measurements (e.g. for enabling the elimination of extreme, erroneous observations [17]).

B.2.6 Assessment of required bandwidth

The goal of analysis is an estimation of resources required to serve the measured traffic. In IP QoS networks, this kind of analysis is performed for example by the measurement-based admission control (MBAC). The MBAC attempts to get knowledge about the current state of the resource use by measuring traffic carried in the network. It should be noted, that the AC algorithm is closely related to the specification of the network service, in particular with characteristics of served traffic and QoS guarantees. Some of the known MBAC algorithms are listed below:

- Measured sum algorithm [10]. The required input data for the algorithm is:
    - Measured (estimated on-line) average rate of aggregate traffic
- Hoeffding bound algorithm [6]. The required input data for the algorithm are:
    - Measured (estimated on-line) average rate of aggregate traffic
    - Peak rates of all running flows
- Brichet/Simonian algorithm [6],[16]. The required input data for the algorithm are:
    - Measured bit rate variance of the aggregate traffic
    - Peak and sustained rates of all running flows
- Methods based on the Chernoff bounds and Many Sources Asymptotic [15]. The required input data for the algorithm is:
    - The distribution of measured bit rate of aggregate traffic

The algorithm calculates either the amount of resources required for serving the running flows with assumed QoS, or the expected packet loss ratio, given the available dedicated resources (bandwidth and buffer). The raw traffic trace should be pre-processed to obtain the required statistical data (average or variance of bit rate), which is then submitted to the main analysis algorithm. In the case of MSA algorithm, the input data must contain the entire distribution of measured rates.

Exemplary tools (more details can be found in [4]):

- http://tracer.ucnet.uoc.gr/interface/src/index.html: Advanced Traffic Analysis Tool, for calculating loss ratio, required resources, admissible region etc., based on traffic traces.

- http://www.statslab.cam.ac.uk/~djw1005/Stats/Lecture/Topics/eb.html: scripts for calculating empirical effective bandwidth from tcpdump trace.

B.2.7 Assessment of traffic descriptor parameters

The aim of the analysis if to fix appropriate parameters of the traffic descriptor. Traffic descriptor does not try to exactly model the traffic arrival process, but rather to provide envelope, which describes its worst-case behavior. In the IP QoS networks, the traffic descriptor is used for specifying the traffic contract between the user and the network. Therefore, its parameters are input to the policing function and admission control decisions.

- Typically, the parameters of Token Bucket mechanism are used as traffic description. Traffic is described properly, when values of the parameters are minimal and still all traffic is considered conforming. The algorithm for assessment of proper values for those parameters takes the trace of packet sizes and arrival times, and produces the “burstiness curve”, i.e. the set of (r,b) pairs which properly characterize the traffic.

- Recently, the Traffic Envelopes have gained attention as an alternative method for traffic description [11]. The algorithm for calculating the envelopes takes as input the trace of packet sizes and arrival times, and produces the traffic envelopes, i.e. set of rates, measured over different intervals.

B.2.8 Traffic matrix estimation

In fact, this is a quite complicated traffic analysis process, requiring common processing of different data sets: measured load on each link and the information on current routing configuration and network topology. The output data is a matrix of estimated traffic demand between each pair of nodes. The outline of the methodological approach, together with some references, can be found in [8].

B.2.9 QoS analysis

This time, the input data consists of a set, or time-series, of measured values of QoS parameters: OWD, IPDV, loss, RTT. The singleton values correspond to the measurement taken on a single packet (probe in the case of active, and real data packet in the case of passive measurements).

- If the measurements were performed over certain time interval, the obtained data set can be treated as a sample of values, which can be analised using the previously mentioned methods for calculation of empirical statistical parameters, and for fitting to known probability distributions.

- If the consecutive obtained values are treated as a time series, it can be analised using prediction methods, mentioned previously.

- Possible analysis includes also mapping of the measured packet level QoS parameters into the subjective Mean Opinion Score (MOS) scale, to assess the quality perceived by the users of voice or video application. In the case of voice, this requires for example applying of the ITU E-model, which allows for translating network-level transfer quality to user-level QoS [18].

- In certain situations, observation of QoS measurement results allows for recognizing of occurrences of specific events (like link failures or route changes). The analysis algorithms, which belong to this class, require applying the data mining techniques, (e.g. pattern detection [18]), to different data sets, like QoS results and routing configuration data.

B.2.10 Traffic analysis for intrusion detection

The goal of the analysis is to search in the trace of observed traffic for abnormal patterns, possibly related with occurrences of typical network attacks.

References

[1] EGSO: European Grid of Solar Observations http://www.mssl.ucl.ac.uk/grid/egso

[2] Mark Allman, Ethan Blanton, Wesley Eddy. "A Scalable System for Sharing Internet Measurements". *Proceedings of the Passive and Active Measurement Workshop*, March 2002

[3] "Metadata Management of Terabyte Datasets from an IP Backbone Network: Experience and Challenges" by Sue B. Moon, Timothy Roscoe, presented at ACM SIGMOD Workshop on Network-Related Data Management, Santa Barbara, CA. May 2001.

[4] C. Schmoll et al., „D11 – State of Interoperability", MOME deliverable, June 2004

[5] P. Tran-Gia, N. Vicari (eds.), Proceedings of the COST257 Mid-Term Seminar, Vilamoura, Portugal, 1999

[6] P. Tran-Gia, N. Vicari (eds.), "Impacts of New Services on the Architecture and Performance of Broadband Networks", COST257 Final Report, compuTEAM, Wuerzburg, 2000

[7] D. Veitch, P. Abry, "A wavelet based joint estimator of the parameters of long-range dependence", IEEE Transactions on Information Theory, vol. 45 no. 3, pp.878-897, 1999

[8] K. Salamatian, S. Fdida, "A framework for interpreting measurement over Internet", Proceedings of the ACM SIGCOMM workshop on models, methods and tools for reproducible network research, Karlsruhe, Germany, 2003

[9] A. Klemm, C. Lindemann, and M. Lohmann, "Modelling IP Traffic Using the Batch Markovian Arrival Process", Performance Evaluation 54, pp.149-173, 2003

[10] L. Breslau, S. Jamin, and S. Shenker, "Comments on the performance of measurement-based admission control algorithms", IEEE INFOCOM'2000, Tel Aviv, Israel, March, 2000

[11] J. Qiu, E. Knightly, "Measurement-based admission control with aggregate traffic envelopes", IEEE/ACM Transactions on Networking, 9(2), pp. 199-210, April 2001

[12] S.Low, P.Varaiya, "A simple theory of traffic and resource allocation in ATM", IEEE GLOBECOM '91, volume 3, December 1991

[13] G.J. Hahn, S.S. Shapiro, "Statistical Models in Engineering", Wiley, 1994

[14] P. Salvador, A. Pacheco, R. Valadas, "Multiscale Fitting Procedure using Markov Modulated Poisson Processes", Telecommunication Systems Journal, 23(1-2):123-148, June 2003

[15] C. Courcoubetis, V.A. Siris, G.D. Stamoulis. "Application of the Many Sources Asymptotic and Effective Bandwidths to Traffic Engineering", Telecommunication Systems, 12(2-3): 167-191, 1999

[16] M. Dabrowski, F. Strohmeier, "Measurement-based Admission Control in the AQUILA Network and Improvements by Passive Measurements", Architectures for the Quality of Service Internet, W. Burakowski, B. Koch, A. Beben (eds.), LNCS 2698, Springer 2003

[17] L. Kovacs, D. Vass, A. Vidacs, “Improving Quality of Service Parameter Prediction with Preliminary Outlier Detection and Elimination”, IPS’2004, Budapest, Hungary, March 2004

[18] I. Miloucheva, A. Nassri, A. Anzaloni, “Automated analysis of network QoS parameters for Voice over IP applications”, IPS’2004, Budapest, Hungary, March 2004

[19] R. Riedi, M. Crouse, V. Ribeiro, R. Baraniuk, A Multifractal Wavelet Model with Application to Network Traffic, IEEE Transactions on Information Theory, Vol.45, No.3, April 1999

[20] EuroNGI: Design and Engineering of the Next Generation Internet -Towards convergent multi-service networks; http://eurongi.enst.fr/en_accueil.html