An Ethernet Segment Performance and Workload Characterization Using Set of Filters Based on Free Software Tools

Vadim A. Ponomarev[†], Dr. Iouri A. Bogoiavlenski[‡], Dr. Timo Alanko[§]

[†,‡]Department of Computer Science, University of Petrozavodsk
[§]Department of Computer Science, University of Helsinki

[†,‡]Lenin St., 33, Petrozavodsk, Republic of Karelia, 185640, Russia
[§]P.O.Box 26 (Teollisuuskatu 23) FIN-00014 University of Helsinki, Finland

E-mail: [†]vadim@kftt.karelia.ru, [‡]ybgv@mainpgu.karelia.ru, [§]alanko@cs.Helsinki.FI

Abstract

The research presented in this paper has two main goals. First, basing on measurements to characterize the workload and the performance of the Ethernet segment, which provides data communication services for the local users of the Federal Petrozavodsk RUNNet Node. Second, to develop software tools, which are flexible enough to be used in analyzing any Ethernet segments.

Characteristics of interest were throughput in bytes and in packets, the utilization of the network, and the distribution of the traffic according to different low–level protocols and to servers connected to the segment. Some other characteristics of the system behavior are also shown.

Measurement techniques and the data storing scheme are specified. The measurement tools, based on the use of the *tcpdump* package, were implemented as a set of filters in the Linux environment.

A two–layer scheme for filtering of LAN packets was developed; the aim was to let the measurement tools work concurrently without loosing packets.

The results of the performance measurements are presented and discussed, and the main classes of the traffic workload are characterized using appropriate graphical visualization.

The research was done in frame of the Joint Research Project [1].

1 Introduction

The Federal Petrozavodsk RUNNet Node (FPRN) has since some time being in the stage of fast growth [2]; its configuration at the time of this performance study is shown in Figure 1. The LAN is one of the most important subsystems of FPRN, it is implemented by the Ethernet (10 Mbps) technology and consists of two segments, called "Core LAN" and "University LAN".

The "University LAN" segment provides data communication services for close to 200 workstations used by the staff and by students, and for two NetWare servers `novell` and `student`. The segment is characterized by a fast growth as everything is increasing: the number of workstations, their capacity, and the range of services provided by the FPRN servers. This obviously leads to a strong increase in the demand of network bandwidth and makes segment capacity planning an acute problem. The first stages in the problem solution are estimation of the current performance through traffic measurements, and characterization of the workload [3]. This paper describes a practical software methodology for the solution of these two problems. It also presents the measurement results concerning the segment performance and the workload characteristics; some discussion is also included.

We consider the "University LAN" segment with its servers and workstations as "the system under test" and the Ethernet transport system as "the component under study". Packet transportation is the main service of this component. Its main performance metrics are throughput (of bytes or of packets) and utilization. The factors that affect the performance are the packet arrival rate and the packet size. In order to get a more detailed insight into the characteristics of the workload we divided it into classes according to the protocols used and according to the servers of the

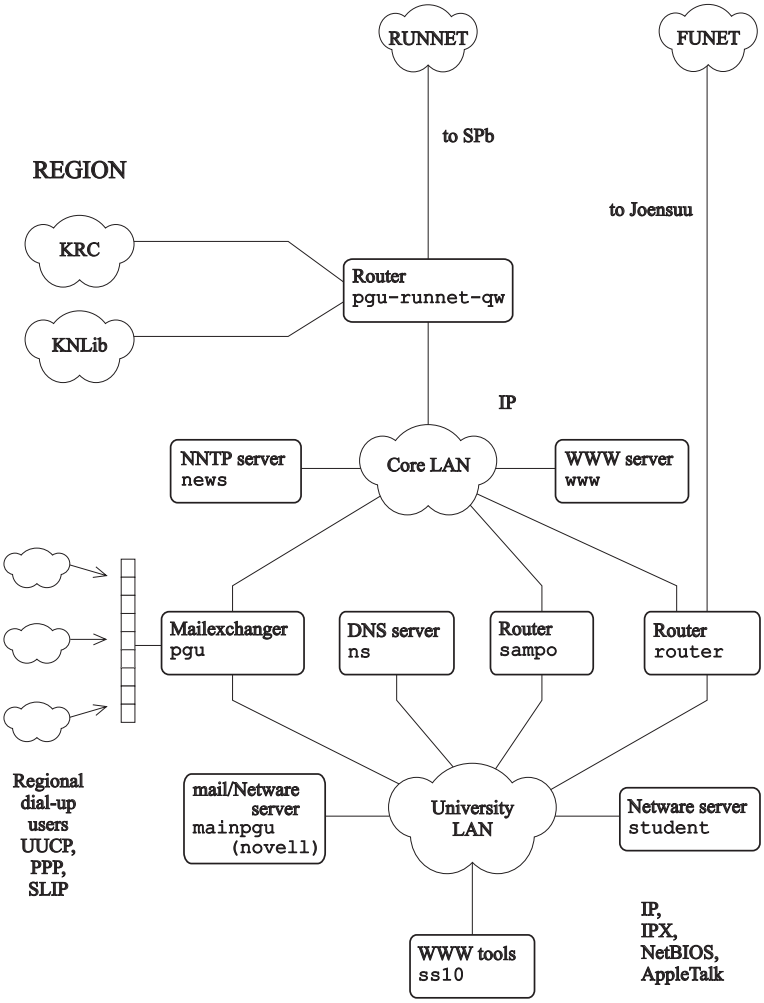

Figure 1. The Federal Petrozavodsk RUNNet Node after March, 1st, 1997

network.

There exists a number of software packages for measuring, collecting, processing, and visualization of traffic data in an Ethernet segment. However, these packages are rather expensive, and as their source code is not available, they are not flexible enough for our purposes. To solve the problems, we had to develop new software tools.

The tools operate in the Linux operating system environment, and they are implemented as a set of filters, based on free software packages. The packages used are *shell*, *tcpdump* [4], *tcp-reduce* [5], *awk* [6], and *gnuplot* [7].

The measurements took place between the 30th of March and the 7th of April in 1997. The observation point was a Linux workstation inside the University LAN segment (see Figure 1).

The analysis of the results shows that during the measurements period the segment was, most of the time, under a rather light load; on the other hand, there were some short periods when the segment was, in practice, close to saturation. A more detailed discussion of the results is presented in the Sections 4 and 5.

The set of filters developed uses a special two–layer packet capturing scheme. The set is easy to use and flexible. It can be used for performance monitoring of an Ethernet segment, and for partitioning the workload into classes according to the various requirements which may appear during the capacity planning process.

The rest of paper is organized as follows. Section 2 specifies the measurements which were done. The set of filters developed is described in Section 3. In Section 4 the measured throughput and utilization of the segment are presented and discussed. At last, in Section 5 the main workload class characteristics are presented and considered.

2 Measurement Specification

Measurements had to provide information to characterize both the stationary behavior of the system and to characterize its dynamic behavior over time. For the latter purpose the time was divided into a sequence of equal intervals; the length of the interval defines data granularity. The time–dependent behavior of throughput and utilization can be obtained counting the number of bytes and packets transferred during each interval.

Choosing of workload classes was based on the following considerations. First, in the segment there are two intensively used NetWare servers: `novell` and `student`; their roles in the segment traffic are of uttermost interest. Second, a substantial part of the workload consists of IP traffic; besides, there seems to be a tendency of application–software migration from IPX to IP and other low–level protocols. Third, the IP fraction of the traffic characterizes users who directly use Internet servers and/or routers of FPRN, located in Core LAN segment. At last, the usage of the TCP/IP application protocols in the IP traffic are definitely of interest.

Thus, at this stage of analysis the workload can be defined as consisting of bytes and packets, and it can be divided into different classes according to the protocol used (IPX, IP, other low level protocol) and the source/destination of the traffic (the two NetWare servers). Specifically, for our investigation purposes the packets to be captured are divided into the following classes:

- all packets
- IP packets only
- IPX all packets
- IPX packets sent from `novell` NetWare server,
- IPX packets sent to `novell` NetWare server,
- IPX packets sent from `student` NetWare server,
- IPX packets sent to `student` NetWare server,
- other IPX packets (not belonging to any of the above IPX classes),
- other packets (not belonging to any of the above classes).

Since the performance of Ethernet is very sensitive to the length of packets [8], the packet length distribution should be evaluated, separately for each class.

Some characteristics common to all traffic are also of interest. The number of active hosts is an important characteristics of user behavior; it is obviously one of the important factors when future loads of the system are predicted. Hence, the number of active hosts is measured. One minute was chosen as the period of host activity registration, and a host was considered active if, during a minute, at least one packet sent from the host was observed.

The traffic of TCP connections was captured and analyzed separately

using the *tcp–reduce* package [5]. For each connection it forms a record containing the following data:

- time when the connection began (first SYN packet)
- duration of the connection in seconds
- applied protocol used in the connection
- bytes sent by originator of the connection
- bytes sent by responder to the connection
- originator host
- responder host
- state that the connection ended in

The output from the measurements is stored on disk for further analysis. A series of preliminary measurements were made with the aim to determine approximately the volume of data to be stored. The experiment showed, that the daily volume of segment traffic was in the range of 12–18 Mpackets. The curves of cumulative sums of transferred packets are shown in Figure 4. They are close to the corresponding curves received during the preliminary measurements.

The first 54 bytes of each packet were stored on disk for our analysis purposes. This means that the amount of disk space needed for storing the daily traffic data was, without timestamps, in the range of 648–972 Mbytes.

However, the disk space available for all the data was only 500 Mbytes. Therefore we decided compress the information "on the fly": we counted the number of bytes and the number of packets transferred during a second, and on the disk we stored a record containing these values and the corresponding timestamp. Eventually, we had a record for each second of a day. Thus, we had to accept an one second–level of data granularity. The data was written into separate files, one for each class.

Notice, that an one–second level of data granularity seems to be quite reasonable, and it is accurate enough for practical purposes. In particular, it has been shown [9] that for the Ethernet traffic the one–second time scale still reflects such an essential feature as the self–similarity. As a matter of fact, the characterization of packet throughput which is based on this data (see Figure 5) is quite similar to the behavior shown in the corresponding pictures in [9], where comparisons of these kinds of pictures are the main argument for the self–similarity feature.

To summarize, for each class three files were written, each containing data of packet lengths, and per-second-level data of bytes and packets transferred.

For each day a new set of files was created. The scheme lead to a rather low volume of data. The size of each packet–size file was close to 1 kbyte. The size of each file containing the byte/packet data was in the range of 750–1200 kbytes. Already the preliminary measurements showed that the size of a file containing the host activity data did not exceed 20 kbytes, and that the size of a file containing the TCP connection records did not exceed 3 Mbytes. Thus the total amount of disk space needed for the daily data did not exceed 25 Mbytes.

The example below, collected during real measurements on the 3rd of April 1997, shows the file sizes in kilobytes:

```
   1  1997-04-03-histo-all
   1  1997-04-03-histo-ip
   1  1997-04-03-histo-ipx-all
   1  1997-04-03-histo-ipx-from-novell
   1  1997-04-03-histo-ipx-from-student
   1  1997-04-03-histo-ipx-other
   1  1997-04-03-histo-ipx-to-novell
   1  1997-04-03-histo-ipx-to-student
   1  1997-04-03-histo-other
  14  1997-04-03-minute-active
1143  1997-04-03-second-bytes-all
1065  1997-04-03-second-bytes-ip
 953  1997-04-03-second-bytes-ipx-all
 885  1997-04-03-second-bytes-ipx-from-novell
 853  1997-04-03-second-bytes-ipx-from-student
 801  1997-04-03-second-bytes-ipx-other
 850  1997-04-03-second-bytes-ipx-to-novell
 823  1997-04-03-second-bytes-ipx-to-student
 874  1997-04-03-second-bytes-other
 964  1997-04-03-second-packets-all
 917  1997-04-03-second-packets-ip
 812  1997-04-03-second-packets-ipx-all
 779  1997-04-03-second-packets-ipx-from-novell
 766  1997-04-03-second-packets-ipx-from-student
 740  1997-04-03-second-packets-ipx-other
 777  1997-04-03-second-packets-ipx-to-novell
```

```
  759   1997-04-03-second-packets-ipx-to-student
  861   1997-04-03-second-packets-other
 2498   1997-04-03-tcp-log
20669   total
```

3 The Set of Filters

3.1 Filters for Traffic Capturing

It was not evident that the capacity of the monitoring node would allow the `tcpdump` to capture the packets without any losses. To find out whether there was enough of capacity, a series of preliminary measurements were made. During the experiments the amount of captured data was compared with the amount of data really transferred. The experiment showed that the standard PC platform with a 66 MHz CPU was not able to capture the packets without losses, but the same platform with a 150 MHz CPU managed the task satisfactorily.

We have developed special filters for producing measurement data files, specified in Section 3. The filter interaction scheme is shown in Figure 2. The scheme has two layers: in this way it is easy to take into account the special features of the monitoring environment discussed in the paragraph above.

The first layer consists of one *tcpdump* process, which is running under the highest possible priority. Only this process is "listening" to the channel. It is tuned to capture the first 54 bytes of each Ethernet packet. The process creates a raw packet stream output, which is redirected through named pipes to three *tcpdump* processes of the second layer; these are running concurrently with normal priorities.

On the second layer three concurrent chains of filters form different measurement–data files, specified in Section 2. Each chain consists of a corresponding *tcpdump* process, which converts the raw packet stream into a readable text form, and of one or more filters producing a subset of data files. The filters are implemented in `awk` and are running under the lowest possible priority.

The first chain calculates and writes on disk files which contain the throughput metrics (packets/bytes per second) for all packets, for the IP packets, and for other packets; the corresponding packet–length distributions are also included. The host–activity data is calculated and written

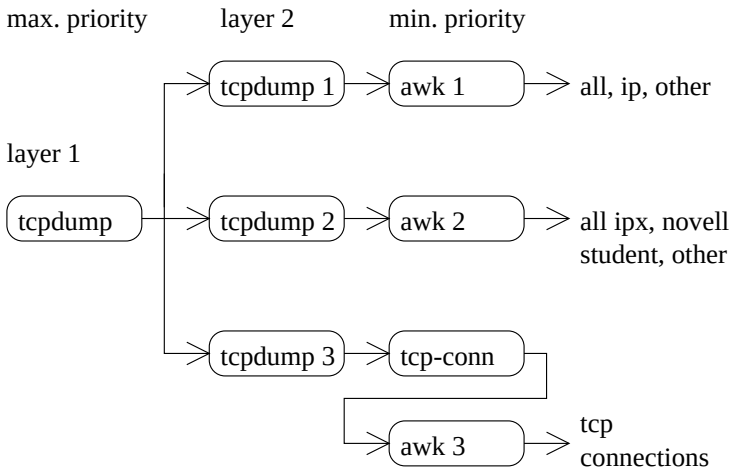

Figure 2. The scheme of interaction between the monitoring filters

on disk. The *tcpdump1* process uses the first 14 bytes of each packet. The `awk1` process then uses a memory array, indexed by the host Ethernet address, to store a boolean value indicating whether the host was active during this minute.

The second chain processes IPX packets only. It stores on disk files the throughput data (packets and bytes per second) for packets sent to/from the NetWare servers, and for packets, which do not belong to these classes. It also stores on a separate file the data needed for the packet length distribution. For this purpose, the *tcpdump2* process uses the first 16 bytes of each packet.

The third chain processes TCP connections. It uses a *tcp-conn* script (from the *tcp-reduce* package) to form the connection records and writes them on disk.

3.2 Filters for Data Processing and Visualization

A set of filters was developed to investigate the statistical properties of the measured data, aggregated on arbitrary intervals of time. The filters

are implemented with *awk* and require as parameters the length of the aggregation interval and the time window of interest.

The measurement data file is received from *stdin*, and the results are written into *stdout*. The statistics calculated are minimum, maximum, mean value, deviation, percentiles, and cumulative sums.

Another set of filters was developed for evaluation of performance metrics and workload characteristics. The filters calculate the needed metrics and provide them in a form suitable for further visualization. In particular, the filters provide throughput and utilization both as a function of time and as histograms.

The visualization filters generate corresponding *gnuplot* programs and then call for them *gnuplot*. Each filter requires as parameters the title of the plot, the time window of interest, and the name of the data file. The filters use a set of auxiliary filters to drop data which do not belong to the time window, to convert the timestamps data from 'HH:MM" into "MMMM" format (to simplify the data used by *gnuplot*), to calculate the minimum and maximum values (for specifying the *yrange* value of the *gnuplot* program), and so on. The plot title, the axes, the horizontal and vertical ranges, the ticks, and the plot itself are generated for each *gnuplot* program. The filters can also generate GIF and EPS files.

The filters were used for a preliminary visualization and analysis only. The main part of the figures presented below were plotted by manually corrected *gnuplot* programs. We have also developed a *bash* script for executing all *gnuplot* programs available in the current directory. Using this script turned out to be both convenient and efficient.

4 Measured Throughput and Utilization

Figures 3 and 4 show cumulative sums of bytes and packets transferred through the segment during 1–4 April 1997. The workloads for 1,2 and 4 April were similar, the total amount of transferred data was close to 4.5 Gbytes, meaning 14 Mpackets per day. The workload for the 3rd of April was heavier: more than 6 Gbytes, 18 Mpackets. For further study we chose the 3rd of April.

For the time interval from 7.30 till 23.00 the curves have a character of an almost linear growth.

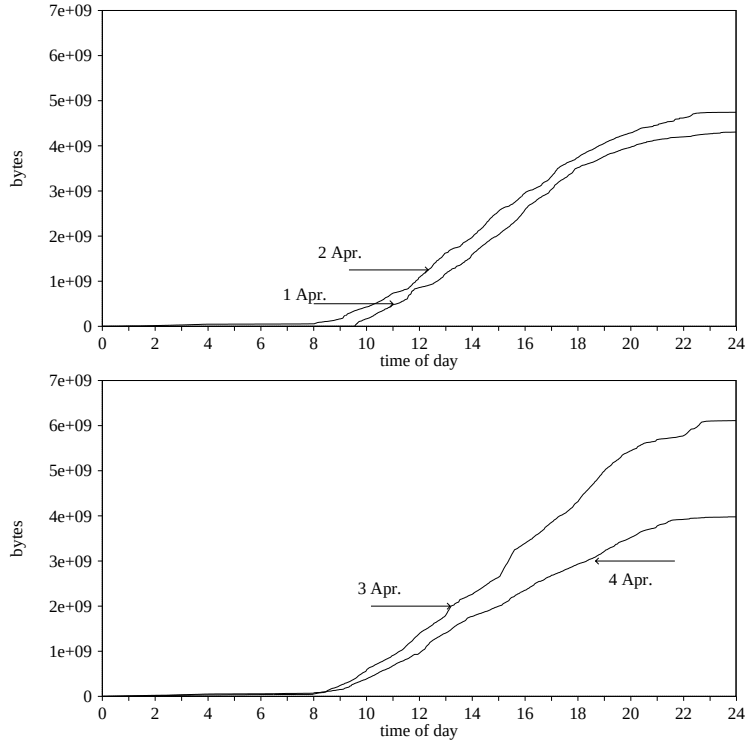

Figure 3. Cumulative sums of bytes transferred

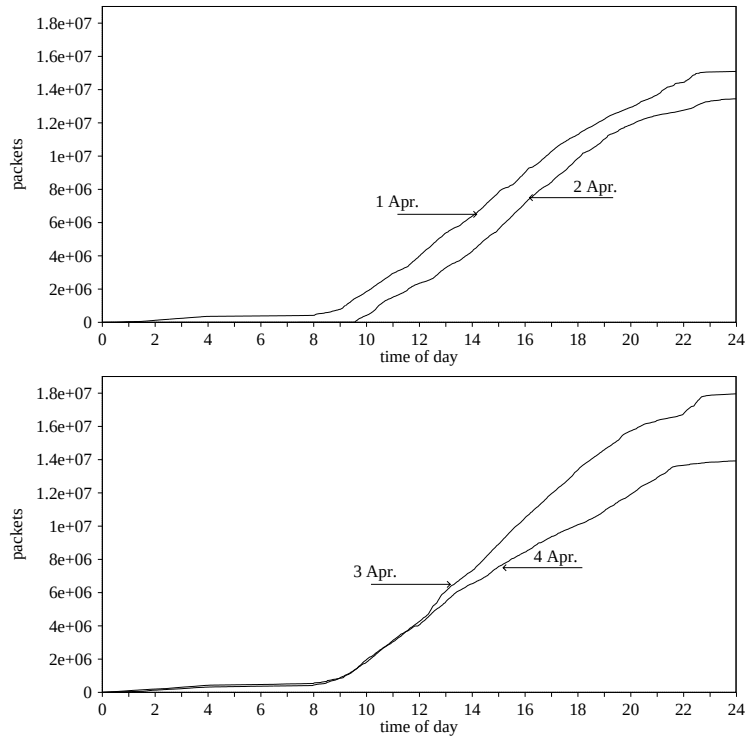

Figure 4. Cumulative sums of packets transferred

Figure 5 shows the throughput of bytes as a function of time. The picture seems to be typical for the Ethernet [9, 10]. The behavior is bursty: intensive periods are separated by less intensive periods. The distribution of throughput is shown in Figure 6. It is worth noticing that for more than 45% of time the throughput, measured in bytes per second, is less than 0.5 Mbps; more than 3 Mbps the throughput is only for 2.6% of the time.

The throughput measured in packets is similar. The reason for differences lies in the packet size distribution, shown in Figure 7: it is obvious that high throughputs in bytes per second occur when large packets are transferred.

The 100–byte peak in the distribution is explained by the existence of a very big number of short control and request messages in the segment traffic. The 600–byte peak of the distribution is mostly due to IP packets and may be explained by the MTU value used in the current TCP/IP software configuration files. The 1500–peak is due to packets sent from the `student` server, and to those exchanged inside the Windows Working Groups (big file transfers).

The segment utilization is computed from measurement data as a ratio of the packet transfer time to the total time. The dynamic behavior of the utilization is shown in Figure 8, and the distribution of utilization is visualized as a histogram in Figure 9. The measurements were based on one–minute measurement periods, and we can notice that at this level of granularity the utilization was less than 0.2 for more than 88 % of time, for 10% of time it was between 0.2 and 0.3, and for 1.7% of time it was between 0.3 and 0.4; at the minute–level it never exceeded 0.4.

As a whole, during the measurement period the LAN was, most of the time, under a rather light load; on the other hand, there were also some short periods when the LAN was, in practice, close to saturation.

The number of active hosts during the day, together with some statistical characteristics, are shown in Figure 10. One can see, that the number oscillates in the range of 75–100, which is less than half of the total number of hosts connected to the segment. This is consistent with the earlier remark that the segment was, most of the time, under a rather light load. The number of active hosts has a rather stable statistical behavior: its standard deviation is essentially less than its average. Hence, the characteristic can be used for purposes of segment performance modeling.

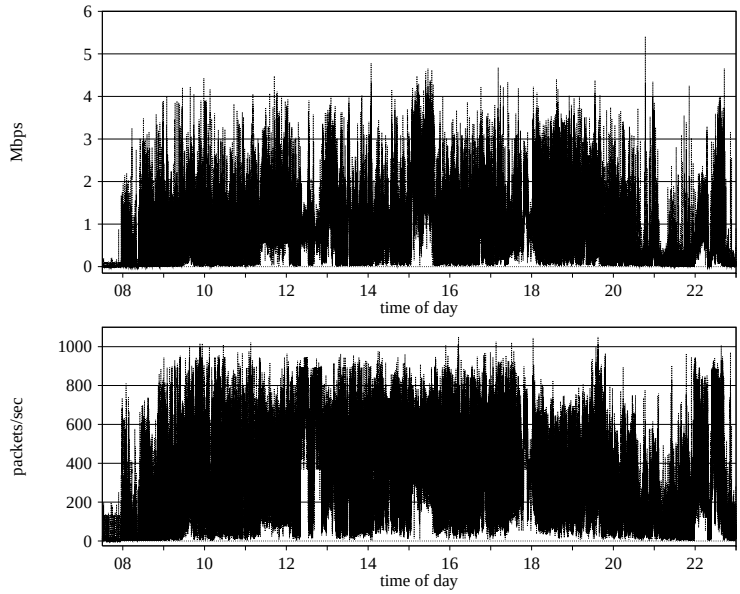

Figure 5. Throughput (bytes and packets)

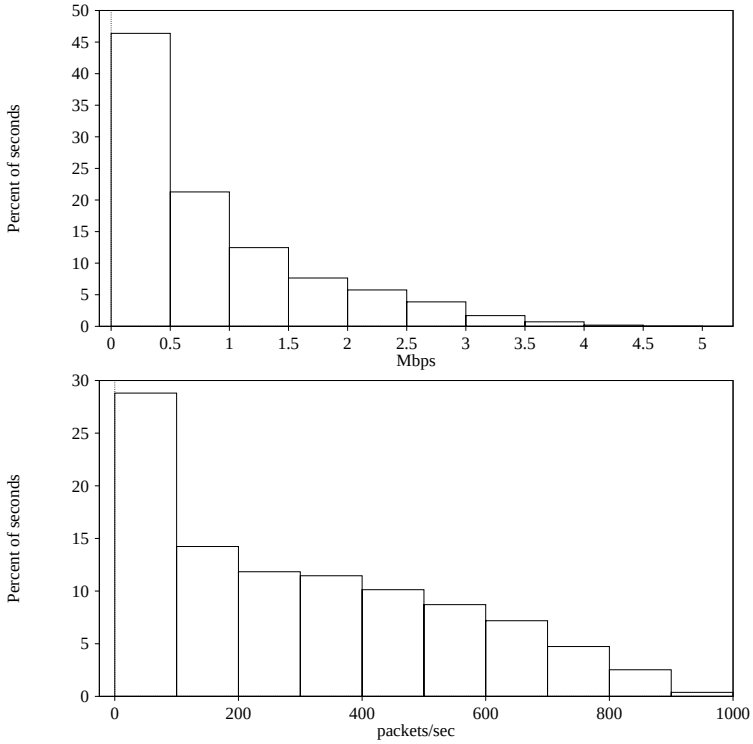

Figure 6. Distribution of throughput

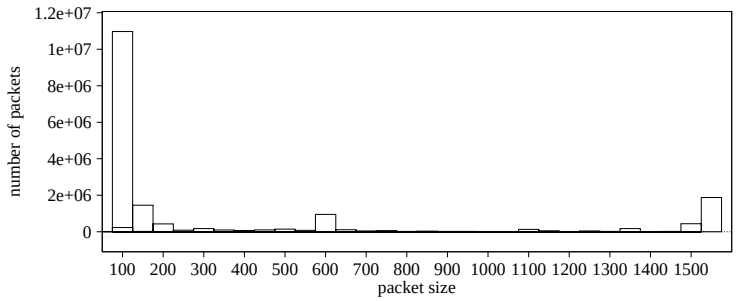

Figure 7. Packet size distribution

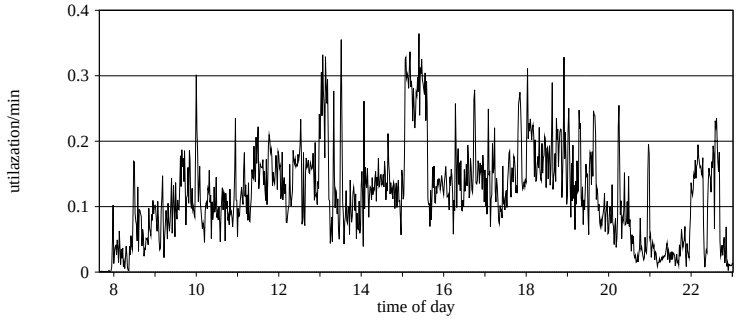

Figure 8. LAN utilization: dynamic behavior

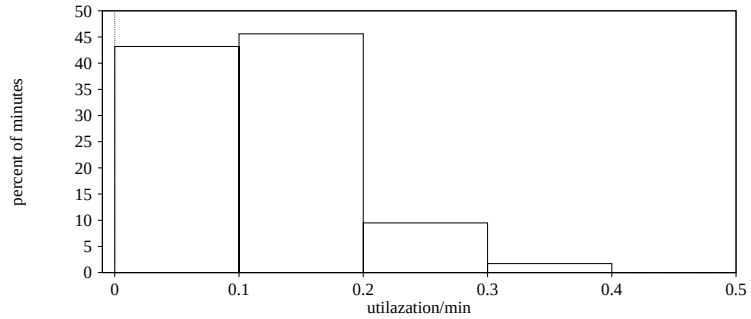

Figure 9. Distribution of utilization

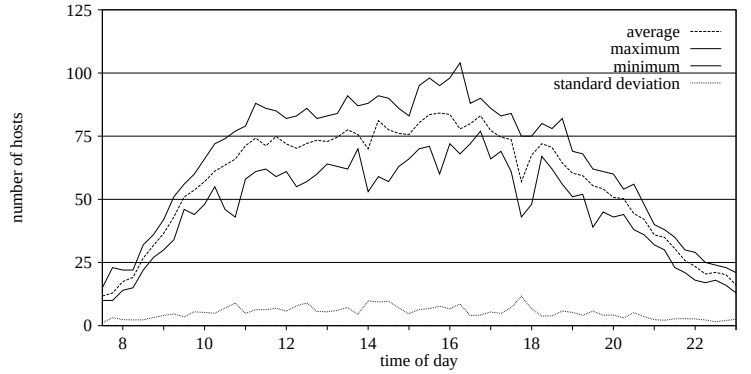

Figure 10. Number of active hosts with corresponding statistics

5 Main Workload Classes

The relative frequencies of data transferred, with respect to the main low–level protocols, are shown in Figure 11. The IPX generates 82%, the IP generates 10%, and the other protocols 8% of the total workload.

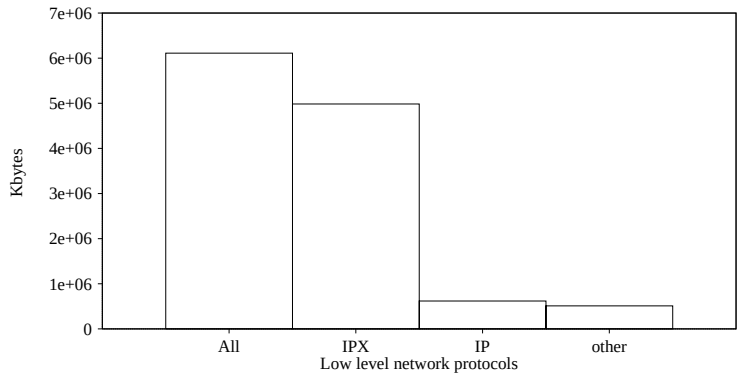

Figure 11. Distribution of packets according to the main network protocols

The IPX traffic (Figure 12) was classified according to the main sources and destinations; these were chosen to be the `novell` server and the `student` server (see Figure 1). The other segment traffic, not related with these servers, was attributed to an aggregated source/destination.

The servers `novell` and `student` were involved in 90% of the traffic, and there was no essential difference between the two. The rest of the traffic was not connected with any NetWare servers. It resulted from data exchange between hosts which are members of Windows Work Groups; in the figure this traffic is denoted as WinWG.

The behavior of the IPX traffic during the day is shown in Figure 13. We notice that the behavior of both servers does not differ very much, and, on the whole, it is also rather stable. The WinWG workload has periodic features: for most of the time the traffic stays at a very low level, but there are two peaks: one around the noon, the other one between 13 and 14 o'clock. In the background there are obviously some irregular user patterns.

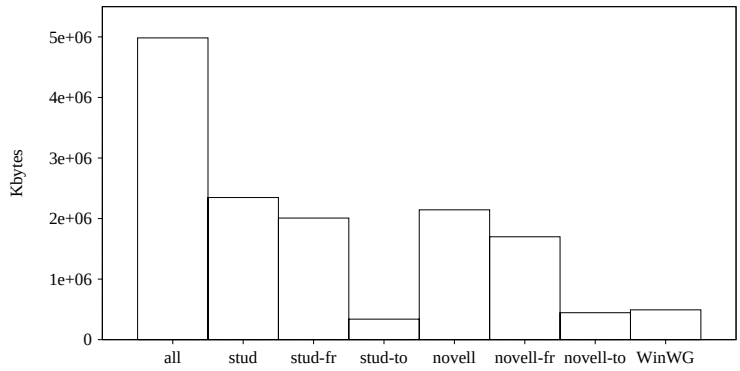

Figure 12. Source–destination frequencies of transferred data

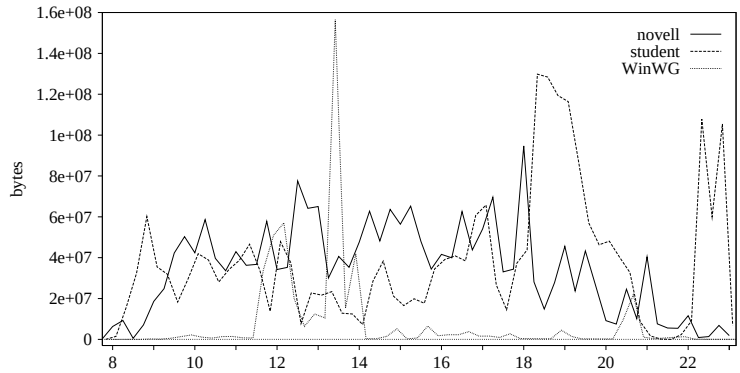

Figure 13. Dynamic behavior of the daily IPX traffic

During 3 April 1997, 34237 TCP connections were identified. Of these 16885 (50%) had a normal SYN/FIN completion. As far as the abnormally terminated connections are considered, in 6460 cases (19%) the program *tcpreduce* could not determine the number of bytes which had been transferred.

The distribution of the TCP traffic according to the application–level protocols is shown in Figure 14. The histogram is based on normally completed SYN/FIN connections and on some RST–completed connections for which *tcpreduce* was able to count the number of bytes transferred. The three main traffic components are due to the applications WWW (37%), proxy (27%), and ftp (20%); together they form 84% of this type of workload.

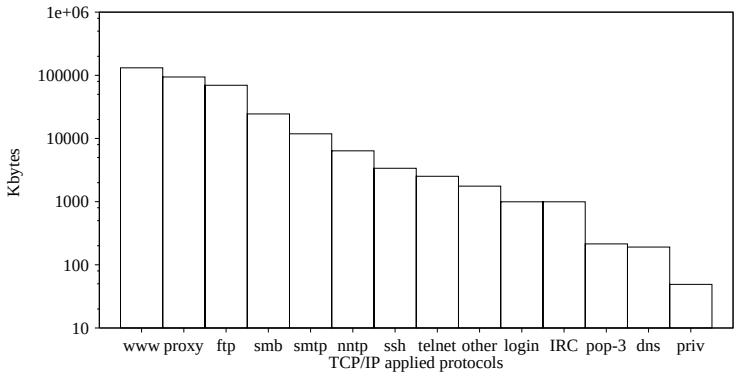

Figure 14. Workload fractions due to the TCP/IP application protocols

6 Summary

In this work a set of filters for the OS Linux environment was developed. The set was based on free software packages, and it allows the evaluation and the visualization of performance and workload characteristics of an Ethernet segment. To avoid packet losses and to collect concurrently all necessary data a two–layer scheme of packet filtering was developed.

Filters of the set are easy and flexible to use. They can be used for monitoring the performance of an Ethernet segment and for partitioning the workload into various classes, always according to the requirements turning up in the capacity planning process.

The set of filters was used to evaluate the performance and workload characteristics of the of the FPRN user segment. Packet delivery was considered as the main service of the segment.

Measurements were specified to evaluate the throughput (in bytes and packets), utilization, and distribution of traffic according to different protocols and servers. In addition, the packet length distributions were evaluated separately for each fraction of traffic, and the number of active hosts was monitored.

The results show that the segment traffic has the following essential features.

The throughput measured in bytes per second is less than 0.5 Mbps for more than 45% of time; it is more than 3 Mbps only for 2.6% of the time.

The packet length distribution has three peaks: at 100, 600 and 1500 bytes. The 100–byte peak is due to the short control and request messages. The 600–byte peak has its origins in IP packets and can be explained by the MTU value used in current TCP/IP software configuration files. The 1500–peak is due to the packets sent from the `student` server, and to those exchanged inside the Windows Working Groups.

The utilization of the segment was less than 0.2 for over 88 % of time; for 10% of time it was between 0.2 and 0.3, and for 1.7% of time it was between 0.3 and 0.4; it never exceeded 0.4.

The usage of low–level protocols was monitored. The IPX generated 82%, the IP generated 10%, and the other protocols 8% of the total workload. The servers `novell` and `student` were involved in 90% of the traffic. The three main components of the TCP/IP traffic workload were due to

the applications WWW (37%), proxy (27%), and ftp (20%); together they formed 84% of this type of workload.

As a whole, during the measurement period the LAN was, most of the time, under a rather light load; on the other hand, there were also some short periods when the LAN was, in practice, close to saturation.

The number of active hosts during the day oscillated from 75 to 100, which is less than half of the total number of hosts connected to the segment. From the management point of view it is satisfactory to notice that the workload generated by these hosts resulted in a tolerable utilization.

Acknowledgments

We want to express our gratitude to Sergei V. Matsko and Alexei M. Minin, the administrators of the Federal Petrozavodsk RUNNet Node, for various consultations and assistance.

Vadim A. Ponomarev wants to thank Markku Kojo and Jarkko Sevanto for all their help and for the knowledge he received during his visit at University of Helsinki in 1996, and for all the discussions and comments at the FDPW'97 in Petrozavodsk and later.

We also want to thank the Solid State Physics Department for the permission to use the host `kftt.karelia.ru` for traffic monitoring.

References

[1] *Timo Alanko, Iouri A. Bogoiavlenski* Performance analysis and capacity planning for a message server in a regional data communication network. Plan of Joint Research. Universities of Helsinki & Petrozavodsk, Departments of Computer Science, 1995

[2] *Victor N. Vasiliev, Natalia S. Ruzanova, Timo Alanko, Iouri A. Bogoiavlenski* An Approach to Capacity Planning of a Local Service Provider as an Element of Internet Infrastructure, in this Proceedings

[3] *Daniel A. Menasce, Virgilio A.F. Almeida, Larry W. Dowdy* Capacity Planning and Performance Modeling: From Mainframes to Client–Server Systems, Prentice–Hall, 1994, ISBN 0-13-035494-5

[4] Tcpdump, a tool for network monitoring and data acquisition
ftp://ftp.ee.lbl.gov/tcpdump.tar.Z

[5] Scripts for Summarizing TCP Connections
http://ita.ee.lbl.gov/html/contrib/tcp-reduce.html

[6] Frequently Asked Questions about the awk programming language
ftp://rtfm.mit.edu/pub/usenet-by-group/comp.lang.awk/faq

[7] The Gnuplot Plotting Utility
ftp://ftp.irisa.fr/pub/mirrors/gnuplot

[8] *David R. Boggs, Jeffrey C. Mogul, Cristopher A. Kent* Measured Capacity of an Ethernet: Myths and Reality, in SIGCOMM'88 SYMPOSIUM Communications Architectures & Protocols, Stanford, California, pp. 222–234, ACM Press, 1988

[9] *Will E. Lealnd, Murad S. Taqqu, Walter Willinger, Daniel V. Wilson* On the Self–Similar Nature of Ethernet Traffic, IEEE/ACM Trans. on Networking, vol.2 N 1, Feb. 1994, pp.1–15

[10] *Walter Willinger, Murad S. Taqqu, Robert Sherman, Daniel V. Wilson* Self–Similarity Through High–Variability: Statistical Analysis of Ethernet LAN Traffic at the Source Level, Proceedings of SIGCOMM'95, pp.100–113