Characterization and Modeling of Network Traffic

Shawky, Ahmed; Bergheim, Hans ; Ragnarsson, Olafur ; Wranty, Andrzej ; Pedersen, Jens Myrup

Published in:
Proceedings of 6th International Computer Engineering Conference (ICENCO)

DOI (link to publication from Publisher):
10.1109/ICENCO.2010.5720429

Publication date:
2011

Document Version
Accepted author manuscript, peer reviewed version

Citation for published version (APA):
Shawky, A., Bergheim, H., Ragnarsson, O., Wranty, A., & Pedersen, J. M. (2011). Characterization and Modeling of Network Traffic. In *Proceedings of 6th International Computer Engineering Conference (ICENCO)* IEEE Press. https://doi.org/10.1109/ICENCO.2010.5720429

Characterization and Modeling of Network Traffic

Ahmed Shawky*, Hans Bergheim*, Olafur Ragnarsson*, Andrzej Wratny* and Jens Pedersen*
*Dept. of Electronic Engineering Networks and Security Section
Aalborg Univeristy, Aalborg 9220, Denmark
Email: {asms, bergheim, olafur, awratny, jens}@es.aau.dk

***Abstract*—This paper attempts to characterize and model backbone network traffic, using a small number of statistics. In order to reduce cost and processing power associated with traffic analysis. The parameters affecting the behavior of network traffic are investigated and the choice is that inter-arrival time, IP addresses, port numbers and transport protocol are the only necessary parameters to model network traffic behavior. In order to recreate this behavior, a complex model is needed which is able to recreate traffic behavior based on a set of statistics calculated from the parameters values. The model investigates the traffic generation mechanisms, and grouping traffic into flows and applications.**

I. Introduction

Internet applications providing different services are rapidly growing now a day leading to a significant increase in the amount of traffic routed on networks [1]. As more and more critical services are depending on being connected, it is vital that the Internet Service Providers (ISPs) have an overview of everything happening on their network. They do this to be able to provide better service to their customers, and to detect errors. Because of this there is now a market for companies that provide these telecommunication providers with diagnostic and traffic analysis reports in order for them to be able to monitor the data traffic on their networks. In order to perform these services, it is necessary to constantly record and analyze data traffic on their clients networks, and provide their clients with a report containing the information the client is looking for. This process often involves capturing the raw data on the link, and storing it for offline analysis. This process requires a lot of resources, equipment and storage space. It would be in these companies best interest to find a way to reduce the amount of equipment and storage space to minimize the economical cost and processing power needed to provide these services. One other factor many ISPs and companies analyzing data traffic often want to know is how it may affect the performance and trends in the network if different applications with different characterization than those most common today will suddenly become more popular.

A. Motivation

A way of lowering cost and time consumption as well as providing a mean to test networks performance could be to use only a few aspects and parameters from the traffic to simulate this traffic behavior. Instead of storing the raw data from the network in a certain period of time one could extract only the most important parameters, which would describe the measured traffic. This could lead eventually to creating a model able to reproduce the correct behavior based on these parameters. It will dramatically decrease the size of needed data bases and computation power involved in the process. Apart from economical issues, connected with the need of huge storage space, such a model would have other benefits. It could be used to perform tests and experiments without touching the existing infrastructure and inside traffic. The model could be introduced with new kind of traffic, e.g. new hypothetical application with set of parameters describing it. This would give an output with possible changes in behavior of the network. This information would be of great importance to the network owner, who will get valuable information about how the network will react. This research aims to create a model that can simulate the behavior of a real network given a set of statistical values as input. Since this can be a very complex task, it is necessary to limit the scope of the research. The key points that this research aims to answer are:

- How to characterize traffic in networks?
- What is the smallest set of parameters needed to characterize this traffic?
- How to collect these parameters?
- How can the traffic be modeled using these parameters?
- How are the results compared to the original traffic?

First it is necessary to find out what type of traffic is flowing through todays networks. The next task is to find out if it is possible to characterize this traffic, using only statistical parameters, and find out which parameters are most critical for correct modeling. Then research will be conducted to find methods on how to collect statistical data from the network. The model will output simulated traffic. If the model is correct, the statistical properties of the simulated traffic should be similar to statistical properties of the original traffic. Fig. 1 shows an overview of how the whole process should work.

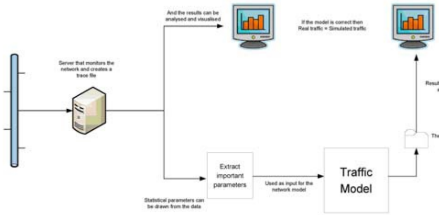

And the results can be analysed and visualised
If the model is correct then Real traffic = Simulated traffic
Server that monitors the network and creates a trace file
Results can be analyzed and visualised
The model produces a tracefile
Extract important parameters
Used as input for the network model
Traffic Model
Statistical parameters can be drawn from the data

Fig. 1. Over View of the Research

II. Traffic Characterization

The traffic is a function of peers connected to the network sending requests to other peers. Peer is defined here as endpoints where a user connects to the network, also called network termination. Each peer has its own IP address. The IP address can therefore be a measure of number of peers connected to the network at a given time. The messages sent between the peers on a network, are split into pieces, called frames, for transportation. When frames traverse the network from one IP to another, they depend on a transport protocol to deliver the frame. Since these protocols work in fundamentally different ways, it is necessary to keep a measure of how these transport protocols are distributed. The applications used on the network uses specific port numbers to be able to communicate, and these can be used to identify which application the frame originated from. Frame size can depend on the application in use, and controls the utilization of the link together with the inter-arrival time. The inter arrival time can be influenced by applications, and is important for the utilization and performance of the network. Utilization is a function of the inter-arrival times and frame sizes. Although its not a direct parameter it is treated in the report as a parameter due to its important characterization effects on networks. As a result it has been decided that the model should be able to regenerate these aspects of traffic behavior:

- Aggregated bandwidth usage
- Frame size distribution
- Protocol distribution
- Application/Port usage distribution
- Inter-arrival time behavior
- IP-address

III. Model Planning

All parameters are to be modeled independently of each other. This means that it is necessary to find out what characterizes the distributions of each parameter, and regenerate this aspect of the traffic. The model design aims in a way to simulate Traffic data behavior rather than mirroring it. It also aims to make it easier to control the models input in order make it possible to simulate different traffic scenarios. To achieve these goals, the model considers the flow concept in the session layer of the network. A flow is a bigger set of data, split into a number of frames, which is being exchanged between two sources within a certain amount of time. A flow is defined here by a 5-tuple containing source and destination IP addresses, source and destination ports and the transport protocol. The frames sharing the same values for these parameters belong to the same flow. Each flow is then assigned to a particular application based on its source and destination ports. Table I shows the applications that the model identifies according to their port numbers.

TABLE I
Application ports used for the model [8!].

Application name	Port Numbers
HTTP	80, 443, 8080
FTP	20, 21
P2P	6881-6999 , 4661-4665
DNS	53
Other TCP	Unlisted ports using TCP protocol
Other UDP	Unlisted ports using UDP protocol
Non IP	99999

A. Modeling Chosen Parameters

Since the model spans several parameters, simple approaches to modeling are used, to get an overview of how the traffic looks like.

1) Modeling Inter-arrival times: A simple approach of finding the inter-arrival process of the network would be to start by checking if the frame arrivals per second in the network could be described by a regular Poisson process. Research suggests that even though some of the complexities in data traffic cant be explained by Poisson when looked at separately, the aggregated traffic might still be Poisson distributed [2]. If the traffic is indeed distributed in a Poisson manner, the inter-arrival times will be independent of each other. It can be checked if this is the case, by finding and plotting the autocorrelation of inter-arrival times. If they are indeed independent, the autocorrelation value should be close to zero. This way it is possible to check if the observed interarrival times on the network is independent, and compare it to the generated traffic. The tail-behavior of the inter-arrival time distribution can be found by plotting the complementary cumulative probability distribution on a log-log scale, and compare the tail of the observed traffic to the generated traffic and other distributions. This will tell whether the observations made on the network are likely to have been generated by an Exponential distribution or if other distribution fits the generated traffic better. The cumulative probability distribution can be plotted and compared to other distributions. The kurtosis and coefficient of variance can also be helpful for explaining how the inter-arrivals are distributed.

2) Modeling IP Addresses: The number of IPs on a network increase in time, as packets from new IP addresses arrive/depart. The numbers of IP addresses going through the link in a certain time period can then be compared to how much traffic there is at that specific time period and then used in the model to make it more accurate. The cumulative development of new IPs from the start of the trace can be plotted as a function of time. This can be done both for the observed and the generated traffic, and the results can be compared.

3) Modeling Transport Protocols: The packets are classified into 3 classes being TCP, UDP and OTHER. Intuitively the protocol of a previous packet influences the probability of the protocol for the next packet. A simple way to model this behavior would be by predicting the probability of the next packet given the current packet. This way a slight correlation between the packages is assumed. The probabilities could then be put up in a matrix, as shown in Table II.

TABLE II
TRANSPORT PROTOCOL PROBABILITY MATRIX.

Previous packet/next packet	TCP	UDP	NONE
TCP	0.7	0.2	0.1
UDP	0.3	0.6	0.1
OTHER	0.6	0.4	0.0

4) Modeling Frame Sizes: By retrieving the frame size of each packet, a discrete probability distribution can be made, by grouping the observations together. The minimum size of an Ethernet frame is 64 bytes. Then the frame sizes can be grouped in different groups according to size up till the maximum frame size of 1518 bytes. The probability of a packet belonging to one group can then be found easily, by looping through the list of all frames, and assign their frame size to one of the groups. The number of frames in a group needs to be normalized with the total number of frames to find the probability. The distribution of observed and generated frame sizes can be plotted in a histogram, and compared. It is also possible to check if the frame size is dependent on other parameters, e.g. protocol distribution by checking the correlation between them.

5) Modeling Port Numbers: The port numbers are used to identify the applications generating the flows. A distribution of port numbers is created and used in the models input as the probability for the type of application of a new flow. While observing traffic on a network we can create a distribution of port numbers. On a certain period of time it is possible to count number of packets destined for a certain port number. It could be very useful to group the measurements into four sets Well-known (0-1023), Registered (1024-49151), Dynamic (49152-65535) or Other (a non-ip packet), to see how they are distributed. Using the knowledge about the source and destination ports of each packet in the trace, a probability distribution can be made in much of the same manner as what was done modeling protocols. Based on whether the last packet is Well-known, Registered, Dynamic or other, the probabilities for the next packet can be calculated. An example of the resulting matrix is shown in Table VII

TABLE III
AN EXAMPLE OF HOW PROBABILITIES FOR PORT NUMBERS ARE SET UP IN THE MODEL

Previous packet/next packet	Well-known	Registered	Dynamic	Other
Well-known	0.4	0.5	0.1	0.0
Registered	0.3	0.2	0.3	0.2
Dynamic	0.4	0.2	0.2	0.2
Othetr	0.6	0.1	0.3	0.0

The resulting distribution can then be seen in Fig. 2

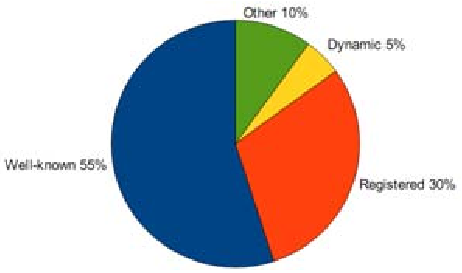


Fig. 2. Port distribution

The distribution of ports can be plotted and compared, to see if they match. It can be interesting to see if the port numbers are correlated in any way with itself or other parameters. This can be checked by plotting the autocorrelation of source and destination ports of the observed and generated frames, to see if the correlations look similar. The correlation coefficient between ports and other parameters can also be checked.

6) Modeling utilization: Considering Ethernet network, the utilization parameter can only take values of 0% or 100%. Fig. 3 shows how the utilization of a link over a one second period could look like.

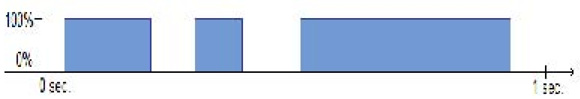


Fig. 3. Example of utilization

To model the utilization parameter information about the number of frames on a link over a certain period of time and the frame sizes are required. A good approach could be to use a frames per second parameter and a frame size parameter. Having distributions of these two parameters it is possible to recreate the utilization of the network. By multiplying the average size of a frame with number of frames per second will give the amount of data (in bytes) to be sent per second and the percentage of the available bandwidth it consumes. The correctness can be measured by measuring the mean and confidence intervals, together with comparing the standard deviations of the processes.

IV. IMPLEMENTAION

The model is realized using two different programs. The two programs are called Analyzer and Generator, respectively according to their tasks.

A. *The analyzer program*

The Purpose of this program is to analyze tracefiles. The program is able to read Extensible Markup Language (XML) files exported from Wireshark [9]. Wireshark is a tool capable of inspecting frames captured from networks and save them into capture files. Capture files are often in the format of PCAP which Wireshark can read. Wireshark is able to export the information from PCAP files into Packet Detail Markup Language (PDML) files which is in XML Format. The analyzer parses the PDML-file made by Wireshark to get the values of the parameters required by the model. The frame number is also read, to verify that all frames have values for these parameters. The values are stored in Python Lists, a versatile data-type in Python. A list contains e.g. the transport protocol for all the frames. There is one list for each parameter. In addition to this, the Analyzer calculates and plots the autocorrelation function for inter-arrival time, frame size, transport protocols, source ports and destination ports. It calculates the correlation between frame sizes and protocols, frame sizes and source ports, frame size and destination ports and the correlation between source and destination ports. It also calculates confidence intervals for the mean of the frames per second process and the utilization process. It can display plots and graphs to help investigate the statistical properties of the trace. Finally the program writes the mentioned statistical values to a file. This file is then used as an input for the Generator program.

B. *The Generator Program*

The generator constantly generates new flows which then create data frames. A new flow starts based on the number of flows per second parameter which is an input value for an exponential distribution. It is decided according to the distribution of flow types which flow model to use. As it is assumed that the original traffic was measured on a directional link new flows are divided into two different directions and treated separately. Flow durations are based on the average duration of flows of the same type, and then picked from the exponential distribution. Each flow is assigned a size in bytes at initialization. It is this size that controls when the flow is terminated. When a new flow has been created it starts to create frames, the arrival times of these frames are computed based on average frame inter-arrival time parameter (which is previously computed based on the average number of frames and average duration time of the flows of this flow type) with use of the exponential distribution. The generator returns an output file that can be used for the analyzer program to create statistics and graphs that can be compared to the results from the original traffic. The process of generating traffic from the model is shown in Fig. 4

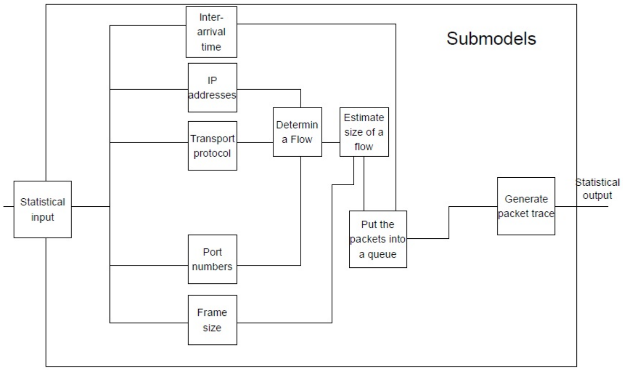


Fig. 4. Overview of the generator

V. RESULTS

These are the results of the model from publically available traces [10]. Three tests were done for two different traces, Trace1 and Trace2. The results are similar, so only the results of one of the tests are presented.

A. *Inter-arrival Times*

TABLE IV
INTER-ARRIVAL TIMES RESULTS

		Original Traffic	Genetraed Traffic
Inter-arrival times	Auto-corr	No Correlation	No Correlation
	Tail Behaviour	Heavy Tailed	Varying arroud exponential

B. *Ip Addresses*

TABLE V
IP ADDRESSES RESULTS

		Original Traffic	Genetraed Traffic
Ip Addresses	Src Address	13509	27105
	Dst Address	21219	32693

C. Frame Size Distribution

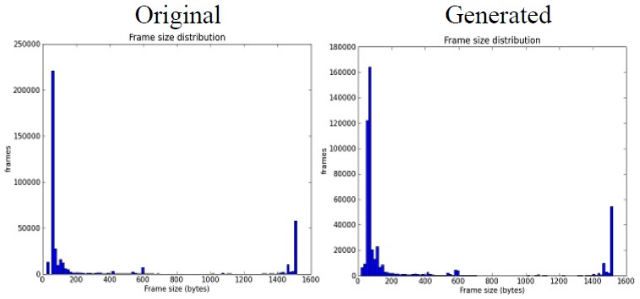


Fig. 5. Frame Size Results

D. Transport protocol

TABLE VI
TRANSPORT PROTOCOL RESULTS

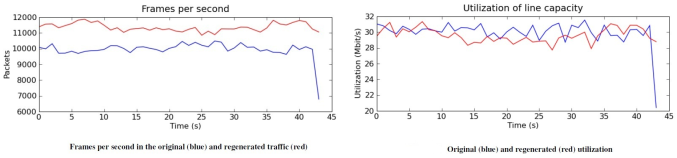
		Original Traffic	Genetraed Traffic
Transport protocol	Auto-corr	Small correlation	No Correlation
	Protocol distribution TCP/UDP/Oth	80.8 / 12.6 / 6.6	79.4 / 13.4 / 6.7

E. Application Distribution

TABLE VII
APPLICATION DISTRIBUTION RESULTS

Application	Original Flow (%)	Generated Flow (%)	Difference
HTTP	22.64	23.57	+ 0.93
FTP	1.292	1.31	+ 0.018
P2P	1.235	1.61	+ 0.375
DNS	22.67	21.96	- 0.71
Other TCP	26.46	26.99	+ 0.53
Other UDP	21.06	20.09	- 0.97
Other Flows	4.61	4.43	- 0.18

F. Utilization



Frames per second in the original (blue) and regenerated traffic (red)

Original (blue) and regenerated (red) utilization

Fig. 6. Utilization Results

VI. CONCLUSIONS

From the results of this research it became clear that a handful of applications are dominating the traffic in backbone networks. It was also made clear that five parameters are considered to be enough to characterize and model the traffic behavior of a backbone networks. The research also showed that using the flow method to model each application separately shows potential, although it is still early in development. This research can be used as a foundation to build a model that is also suitable for simulation of traffic scenarios.

VII. FUTURE WORK

Listed Below are some areas where improvement can be done:

- Improving extraction methods for parameters
- Automate stationarity detection
- Improving queuing system
- Improve flow and application detection
- Add more applications
- Deeper investigation of TCP protocol

REFERENCES

[1] *Global IP Traffic Forecast and Methodology. 2006 20011.* Cisco, 2008.

[2] Karagiannis, Molle and Faloutsos.*A Nonstationary Poisson View of Internet Traffic.* University of California, San Diego, 2005.

[3] Zhang, Xie and Tang.*Bias correction for the least squares estimator of Weibull shape parameter with complete and censored data.* Elsevier Ltd. 2006.

[4] Bernardl and Bosi-Levenbach.*The plotting of observations on probability paper* Stat Neerl. 1953.

[5] Mark Crovella.*Network Traffic Modeling.* Boston University Computer Science. 2004.

[6] Heather Osterloh.*IP Routing Primer Plus.* Sams Publishing. 2002.

[7] Andreas Willing.*A Short Introduction to Queueing Theory.* Technical University Berlin. 1999.

[8] *http://www.iana.org/assignments/port-numbers* . Viewed (1.11.2010).

[9] *http://www.wireshark.org* Viewed (1.11.2010).

[10] *http://tracer.csl.sony.co.jp* Viewed (1.11.2010).

[11] *http://www.itl.nist.gov/div898/handbook/* Viewed (1.11.2010).