Report on WWW-traffic modeling
and WaveLAN measurements

Tapani Nieminen

Helsinki University of Technology
Communications Laboratory
P.O.BOX 3000, FIN-02015 HUT
email: Tapani.Nieminen@hut.fi

1. THE GOAL OF MEASUREMENTS

The basic idea of these measurements has been to get a better understanding of the TCP/IP traffic caused by an individual WWW-user. The measured statistics are aimed to be used in planning of mobile user interface for Internet. The primary issue has been to get parameters for simulation, but also other areas like radio network planning could utilize these results. When the project we found from literature plenty of statistics of WWW-traffic, but they were collected from the WWW-servers or main trunks and intended to optimize their capacity by using a cache and such purposes. During the project there has become article [1], which models network traffic on HTTP level.

2. WHAT WAS MEASURED

The data referred as “old” was got from the measurements done during 5.8. - 23.9.1996 in Helsinki University of Technology from the Communications Laboratory internal Ethernet LAN. The LAN was divided into three segments and there were about 40 PC’s and 5 HP 700-series workstations. The measurement point was in the middle segment, where all the data coming in or going out from the LAN went through. From the same segment the laboratory LAN was connected to the department LAN and farther to Internet, through a PC-based KarlBridge.

The data referred as “new” was got from the measurements done during 6.10. - 11.10.1997 in Helsinki University of Technology from the Communications Laboratory internal Ethernet LAN. The measurement point was in the E-wing segment and there were 19 PC’s and one HP 700-series workstation and a WaveLan base station. Two portable PCs were connected to the base station using wireless LAN product called WaveLan. From the measuring PC we could see all the data moving in the LAN segment. A public domain program called GOBBLER collected the data. In other respects the measurement setup was the same as described in the earlier report [2].

2.1 The setup used with WWW

The setup used in 1996 measurements is reported in [2]. In 1997 measurements the PCs were using the TCP/IP-stack built in to their Windows 95 and Windows NT operating systems. Windows 95 defaults set mtu (maximum transfer unit = the size of largest IP-packet) to be 1500 and rwin (receive window size) to be 8192. Netscape 3.0x and 4.0x were used as WWW-browsers. The options of Netscape limit typically

[1] Mah, B, An Empirical Model of HTTP Network Traffic, Proceedings of INFOCOM’97, Kobe, Japan, April 7-11 1997.

[2] Nieminen, T., Report on WWW-traffic measurements, Helsinki University of Technology, Communications Laboratory, Technical Report T39, 28. October 1996.

the number of (simultaneous TCP-) connections to four, network buffer size to 6 kB, memory cache to 600 kB and disk cache to 5000 kB. If the same page is viewed several times, it is read from the cache. The novelty of the page is verified once during a WWW-session and always, when the user selects to reload the page. With the WaveLan and the reference measurements the cache was not used, so the size was set to 0.

2.2 Data Preprocessing

The data was preprocessed in a HP workstation by two C-language program “tnfiglan” and “tnwwwma7”. In the first one the 1-5 consecutive data files covering 1 - 7 days each were concatenated to form a week long period. All non-WWW packets were filtered out and data parts were cut off leaving only timing information, and MAC-, IP- and TCP-headers, that included all the information needed for the statistics. Then the data of each packet was saved in one of 56 files on the basis of the terminal side MAC address. The WWW-terminal side was chosen on the base of TCP-port numbers. The WWW-server is assumed always to use TCP-port number 80 (0050H) and the other port numbers are assumed to belong to a terminal.

The MAC- and IP-addresses got for the terminals in the measurements were compared to the known addresses of the PCs and HP-workstations in the LAN and portable PCs with WaveLan connection to guarantee, that all the terminals processed really belonged to the Communications Laboratory LAN. To confirm the processing all the exceptions were collected to an extra file and they were identified to some other known equipment, like the ISDN-router. They were excluded from the measuring data, because their usage was small and statistics were assumed to differ in some respect, like delay, from the analyzed groups (PCs and WSs). As result from “tnfiglan” we got for every week 21-32 files, each holding the headers of all WWW- packets coming from (labeled as Uplink) or going to (Downlink) a specified terminal, which had been active during the week.

In the next phase this raw material was collected and changed to MATLAB files by program “tnwwwma7”, which was an enhanced version of the earlier used program “tnwwwmat”. The data was handled in two batches first 46 PC-files and then 2 WaveLan-files, because their statistics were assumed to differ in some respects. The non-zero files are analyzed one at a time so that the statistics of a week actually form a queue of N times one-week terminal sessions. For the MATLAB processing “tnwwwma7” calculates 22 matrices, that can be divided into two groups. They are divided into “All”, “Uplink” and “Downlink” data represent by the second letter “a”, “u” or “d” in the name of the data file.

1. Packet level data are saved in large 7 matrices, which include
- length of every IP-packet as got from the IP-header (“paby.mat")
- length of every IP-packet’s IP- and TCP-headers in bytes ("pahe.mat")
- time label of each packet ("pati.mat")
- number of the acknowledged (or retransmitted) packet, when the TPC has acknowledged new bytes (or retransmitted some) ("pack.mat")
- type of each packet one byte including direction, errors in TCP-mechanism and six status bits from TCP ("ptyp.mat")
- number of the TPC-connection where the packet belongs ("ptcp.mat")
- number of the packet on its TPC-connection ("ppac.mat")

2. TCP-session level data and statistics are saved in 15 matrices, which include

- number of packets in a TCP-connection ("tupa.mat" and "tdpa.mat")
- sum of bytes in a TCP-connection (“tuby.mat" and "tdby.mat")
- sum of bytes in the IP- and TCP-headers in a TCP-connection (“tuhe.mat" and "tdhe.mat")
- starting and ending time of a TCP-connection ("tstr.mat" and “tlst.mat”).
- number of packets and bytes retransmitted in a TCP-connection ("trep.mat" and "treb.mat")
- the host IP-address and the terminal TCP-port number of the TCP-connection ("taip.mat" and “ttcp.mat”)
- the values that were offered for maximum transfer unit in handshaking, when connection is formed (“tumt.mat" and "tdmt.mat")
- status information of the TCP-connections phases gone through and errors if noticed ("pact.mat")

The HTML-protocol can and often also does open separate TCP-connection for different WWW-items like HTML text page, pictures or other such elements. So it is quite usual that there are more than one TCP-connection open simultaneously.

2.3 MATLAB-processing

MATLAB was used to calculate histograms and unite the statistics from different weeks and plot the pictures and graphs. The limitations in available memory and processing capacity forced to calculate the histograms and other statistics for one week at the time. Combining these results created the data for the whole period. This makes it also possible to compare the results between different weeks and/or different terminals or terminal groups.

Each histogram was calculated separately from each terminal and each weekly data and saved in a file with histogram name and suffix “.mvt” to separate them from the input MATLAB-files, which use suffix “.mat”. To be able to calculate the mean and the standard deviation jointly over seven variable size data matrices, three variables, yn, ys and y2, were counted and saved in the same file with corresponding histograms. They include the number of items (yn), sum of items (ys) and sum of squares of items (y2). Calculating data for terminals was done by a program “tn5sum7n” with help of about fifteen smaller programs. “tn5anal” calculated burst and TCP-connection statistics and most of the packet statistics. “tn5anal2” reordered the packet data according TCP-connections and calculated WWW-item, WWW-page and WWW–session statistics and statistics for packet timing inside items. As newer it also calculated statistics for “mini bursts”, I called nibbles. “tn5anar” calculated statistics for response times in acknowledgements and retransmissions. There were almost 200 figures drawn by program “tn5epsp” and saved to postscript files with suffix “.eps”. With the same program similar figures can be produced also from data files grouped by terminals or weeks. But due the limited space, only the most interesting and important figures were picked to this report.

The results have been analyzed on seven logical levels (Packet, Nibble, Burst, TCP-connection, WWW-item, WWW-page and WWW-session). The numerical data has been presented in sets of 186 (42+6*24) figures, which show the distributions calculated from the data. Each figure (fig.) is drawn by MATLAB to fixed size (210 mm x 160 mm) to A4 page and saved in encapsulated postscript file. Here are short descriptions, what they include.

1. Packet level statistics identified with letter ‘p’ are presented in 42 figures, which include

- size of every IP-packet and their data parts total and both directions as got from the IP-headers (10 figures)
- delay from previous packets in same or either link total and both directions in three time scales (15 figures)
- comparisons of delay distributions (6 figures)
- delay from previous packet in the same TCP-connection (1 figures)
- time to acknowledge or retransmit packets in both directions (4 figures)
- delay between packets that belong the same WWW-item, when the direction of the link is model as two-state process (4 figures)
- amounts of WWW-items/page and WWW-pages/session (2 figures)

2. Statistics of higher level logical structures formed from sequences of packets. There are six such groups called Bursts, Nibbles, WWW-items, TCP-connections, WWW-pages and WWW-sessions and all of them are presented in 24 figures, which are named and numbered in systematic order.

- size of groups in packets (3 figures)
- delay from previous group (1 figures)
- length of the group (3 figures)
- cumulative distributions of bytes and average bitrates based on length of group (6 figures)
- distributions of average bitrates of group (4 figures)
- distributions of bytes based on length of group (2 figures)
- size of groups in bytes (5 figures)

I have used the following definitions, when calculating the statistics.

Packet IP-packet, smallest unit of data transmitted on an IP-connection.

Nibble Smallest unit of data to be handled in interworking It is formed from a single packet or a group of packets separated with idle periods of less than 10 ms..

Burst An period of active data transmission, which is defined include only idle periods of less than 2 s. Bursts are separated from each other with idle periods of equal or larger than 2s.

WWW-item A single request/response pair transferring one entity like text page, picture etc. It forms a whole WWW-page or a part of it. It is considered always to be on single TCP-connection and duration is defined to be from the first data packet of the request to the last data packet of the response. WWW-items can not be overlapping in the same TCP-connection, but on other TCP-connections the can be and often also are simultaneous WWW-items.

TCP-connection A numbered connection between WWW-server and WWW-client. It is formed from synchronization handshaking, one or several transfers of a WWW-item and closing handshaking. The older pagers opened a separate TCP-connection for each WWW-item, but presently one TCP-connection can carry tens of WWW-items.

WWW-page	A single WWW-item or a combination of WWW-items, that forms one visual display unit. They are separated by a reading period, which is defined to be from 1 to 300 seconds. If WWW-items do overlap or the time gap between them is less than one second, they are considered to belong to the same WWW-page.
WWW-session	A period during which the client has been actively reading WWW-pages or wait waiting them to be downloaded. They are separated by periods of inactivity, when no WWW-page is downloaded during 5 minutes (300s). The limit tries to reflect the time how long the user can be assumed to actively been using WWW without interrupts, and when an interrupt can be assumed. WWW-client does not send any information when it is closed and a user may keep the client program open indefinitely.

Some of these are just simple and clear definitions for logical phenomena like Packet, WWW-item and TCP-connection and they should be easily calculated out of the available data. Also WWW-page and WWW-session are in principle clearly definable, but since we have not been able to record direct user actions like clicking a page or starting to do something else, we have just used some ad hoc defined time limits to separate them.

Nibble and Burst are just two names and definitions for dividing data into suitable categories the burstyness of traffic patterns, which is the noticeable result from all the mechanisms affecting the connection. The limits are just reflecting the time scale we want to look at them. Idle period of two seconds might be practical to GSM, where bandwidth is narrow and connection should be released for other users to utilize it during idle periods. Actually it takes over a second from a 9,6 kbit/s line to transfer a maximum size IP-packet. I used Nibbles to show that data packets are often forming small groups, which could be handled as one entity in interworking. The 10 ms idle period I selected according the frame size of W-CDMA proposed to be used in UMTS. A 1,2 Mbit/s bandwidth will be needed to transfer a maximum size IP-packet in that time.

The HTML-protocol can have several TCP-connections open and WWW-items downloading simultaneously. So the delay between TCP-connections and WWW-items was just defined, as I felt most suitable. With TCP-connections the delay is measured from the last beginning of a TCP-connection before the present one. Since WWW-items are used in my modeling, I tried to emulate better the protocol mechanism. If possible the delay will be measured from the end of previous WWW-item, which can be assumed to have tricked the next one or released a TCP-connection for it in case maximum number of them were in use. In some cases the first WWW-item seems to trick new ones already before ending and then the delay is counted from the beginning of that WWW-page.

A WWW-session was defined as a period of activity on the connection, during which the maximum delay between two consecutive packets does not exceed five minutes. It was estimated to represent the period, when the user is actively using WWW. When there is no traffic in five minutes the user can be assumed to have closed, put to background or simply forgotten the WWW-session at least for a while. When the length of active WWW-sessions was analyzed in one later lost WWW-reference the differences caused by selecting time limit to 5, 10, 20 minutes were classified to be negligible. It is obvious that no lower level statistics are affected, but in my analysis I could see that about a half of delays between WWW-sessions were from 300 to 1800 seconds (5 minutes to 30 minutes). Such a change in the definition would clearly affect the distributions of delays between WWW-sessions, WWW-session lengths and the number of WWW-pages per WWW-session. The mean of the last one would obviously be doubled.

2.4 Evaluation of the results

The measured data covers WWW-traffic from the whole Communications Laboratory for a period of 7 weeks in 1996 (old data) and a third of laboratory for week in 1997 (new data). The old measurement data covers 48 and the new 6 days. Thirty-two PC-terminals used WWW during the first measurement and nineteen PCs and two portables (WaveLan data, WLAN) during the second. Totally 173 weekly terminal files were recorded in the first and 21 in the second measurement. The data from the two PCs used as reference to WLAN measurements have been analyzed separately (marked LAN). The LAN is included in the NEW, but WLAN is not due it's different delay behavior.

The gathered data amount is rather large. Still this almost a gigabyte of data from eight weeks would flow through an Ethernet at maximum (10 Mbit/s) speed in 15 minutes. Here are the main numbers describing it.

	OLD	NEW	LAN	WLAN
Packets up	1211489	96356	26129	23027
Packets down	1357693	100051	27700	24051
Data-Packets up	125769	15219	4799	4541
Data-Packets down	1130304	80057	23143	19454
IP-bytes up [kB]	81016	8092	2308	2263
IP-bytes down [kB]	756067	76937	21312	17878
Data-bytes up [kB]	32067	4196	1255	1335
Data-bytes down [kB]	701344	72903	20198	16910
Bursts	65053	5418	897	1153
Nibbles	1225740	68728	16919	22909
WWW-items	117085	14469	4682	4411
TCP-connections	114230	9265	1764	1676
WWW-pages	37913	3271	739	780
WWW-sessions	2930	203	9	8
Burst time [s]	178135.00	10325.60	2617.88	2979.93
Nibble time [s]	3593.22	253.862	77.77	91.95
WWW-item time [s]	729463.00	35936.50	7194.38	8113.24
TCP-connection time [s]	6465180.00	274469.00	50157.20	52505.70
WWW-page time [s]	611114.00	19088.80	3169.54	4086.67
WWW-session time [s]	1534020.00	99716.80	11985.60	13136.1
Hosts per WWW-session	3.39	4.48	10.50	11.40

Table 2.1. The main statistics of data measured Packets, IP-bytes and Data-bytes, the numbers and total lengths of Bursts, Nibbles, TCP-connections, WWW-items, WWW-pages and WWW-sessions.

The whole analysis except the packet starting time that was recorded by GOBBLER-program is based on the information included in the headers of IP- and TCP-headers of each packet. So in the figures bytes marked IP-bytes means the bytes included in the IP-packets (headers and data). The size of MAC-packets is with Ethernet 14 bytes more or 64 byte at the minimum. In ATM using LAN Emulation, the MAC-header is 16 bytes, AAL5-header is 8 bytes and padding 0 ... 47 bytes. The ATM headers add five bytes for every 48-byte cell. So the bytes for IP-packet length N bytes will become NA = 53 * (int ((N + 71) / 48)), where int () means taking integer part.

The times are counted as differences between events. In the original GOBBLER files the time passed from the beginning of the measurement is saved into four bytes. Since the time resolution is one microsecond, the counter goes around about 20 times per day. In the preprocessing the fifth byte was calculated on the basis, that records are in FIFO-order. So every time, when the four LSBs had a value, that was smaller than the previous one, the timer was assumed to have gone around and the MSB was increased. With this method from the long quiet periods we would get only the dividend from the period divided by 1 hour 12 minutes counter period. Because this cannot happen very often, its influence on actual results is negligible. Also the times for last events in weekly files are only once under 575 000 seconds, when a week is 604 800 seconds.

On the other hand there are few examples in the old data were possibly rounding has caused extra 4295-second (1 h 12 min) increment to clock during WWW-page downloading. These are partial explanation to over twenty fold values for WWW-item, WWW-page and TCP-connection times, when comparing new data to the old, when the relation in IP-bytes is only tenfold. This does not affect to Burst times, where old in 17,5 times the new, so most of this delay was still caused by older protocol versions with no keep-alive on TCP-connections and smaller maximum packet and receive window size.

The last line in Table 2.1 tells the average of how many WWW-servers the user has visited during a WWW-session. It describes the mobility of traffic, which makes it very difficult get full statistics of a user's WWW-traffic behavior by analyzing WWW-servers logs like for example in [3]. If users on average visit 3 – 5 WWW-servers during a session, the distributions of for example interarrival distributions and traffic density will be misleading when analyzed only from one server.

3. ETSI NRT (WEB) TRAFFIC MODEL AND SOME COMMENTS

Here follows the comparison of the measured data to the model for non-real time services referred later as ETSI model [4]. It is presented in D-ETR SMG-50402 v0.9.3: 4/1997, Annex 2, Paragraph 1.2.2. Traffic models. The definition of the ETSI model is copied and commented in following three pages. The comments of the author use a different font so they should be easily separated from the original text.

Non-real time services

Figure 1.0 depicts a typical WWW browsing **session**, which consists of a sequence of **packet calls**. The user initiates a packet call when requesting an information entity. During a packet call several **packets** may be generated, which means that the packet call constitutes of a bursty sequence of

[3] Aldén M., Traffic Models for WWW User-Behaviour from the Pseudo-Source Level, Telia Research AB, Technical Report 13/0363-04/FCPA 109 0001, 23. May 1997.

[4] D-ETR SMG-50402 v0.9.3: 4/1997, Annex 2,

packets, see [ref 1] and [ref 2]. It is very important to take this phenomenon into account in the traffic model. The burstyness during the packet call is a characteristic feature of packet transmission in the fixed network.

Figure 1.0. Typical characteristic of a packet service session.

A packet service session contains one or several packet calls depending on the application. For example in a WWW browsing session a packet call corresponds the downloading of a WWW document. After the document is entirely arrived to the terminal, the user is consuming certain amount of time for studying the information. This time interval is called **reading time**. It is also possible that the session contains only one packet call. In fact this is the case for a file transfer (FTP). Hence, the following must be modeled in order to catch the typical behaviour described in Figure 1.:

- Session arrival process
- Number of packet calls per session, N_{pc}
- Reading time between packet calls, D_{pc}
- Number of datagrams within a packet call**,** N_d
- Inter arrival time between datagrams (within a packet call) D_d
- Size of a datagram**,** S_d

Note that the session length is modelled implicitly by the number of events during the session.

> The principle of dividing the model to few rather simple layers like session, packet call and a packet called datagram is very good. It describes the quite closely the actual process (here WWW browsing), which generates the traffic. The major drawback in the presented model is the systematic usage of selected statistic distributions. In some cases this is masking out some typical features caused by the used protocols. Good examples of this are the datagram size distribution and average interarrival time distributions. These oversimplifications can make the model misleading, when we try to estimate and optimize things like enveloping and error correction methods for the radio link.

Next it will be described how these six different events are modelled. The geometrical distribution is used (discrete representation of the exponential distribution), since the simulations are using discrete time scale.

Session arrival process: How do session arrive to the system. The arrival of session set-ups to the network is modelled as a Poisson process. For each service there is a separate process. It is important to note that this process for each service only generates the time instants when service calls begin and it has *nothing to do with call termination.*

The ***Session arrival process*** is defined to be a Poisson process. It is natural, that there is a separate process for each service. In most cases they probably could be considered also as independent. No numerical values are presented for any services like WWW browsing, which was mentioned as an example.

The number of packet call requests per session, N_{pc}***:*** This is a geometrically distributed random variable with a mean ?$_{Npc}$ [packet calls], i.e.,

$$N_{pc} \in Geom(\mu_{Npc})\,.$$

The reading time between two consecutive packet call requests in a session, D_{pc}***: :*** This is a geometrically distributed random variable with a mean ?$_{Dpc}$ [model time steps], i.e.,

$$D_{pc} \in Geom(\mu_{Dpc}) .$$

Note that the reading time starts when the last packet of the packet call is completely received by the user. The reading time ends when the user makes a request for the next packet call.

The number of packets in a packet call, N_d: *The traffic model should be able to catch the various characteristic features possible in the future UMTS traffic. For this reason different statistical distributions can be used to generate the number of packets.* For example $\boldsymbol{N_d}$ can be geometrically distributed random variable with a mean *?$_{Nd}$* [packet], i.e.,

$$N_d \in Geom(\mu_{Nd}) .$$

It must be possible to select the statistical distributions that describes best the traffic case under study should be selected. An extreme case would be that the packet call contains a single large packet.

The nature of mechanisms like WWW and FTP is actually to transfer the file or files wanted by the client to the client. This is the basic cause of the active traffic periods, which are named in the ETSI model as ***packet calls***. The used transfer protocol divides the data into packets according its parameters like maximum packet size. Basically the data size of a packet call is a random variable, which in several sources has claimed to be Pareto distributed [4]. The actual packet sizes have quite definite values like 40 or 44 bytes to the non-data maintenance packets and the set maximum size of 552, 576, 1024 or 1500 bytes to the data packets. So I would prefer selecting first a random value for packet call size (file size) and calculating the packet numbers and sizes from that according the used protocol parameters. This would also give more possibilities to test the effects of different protocol parameters to the traffic behavior.

An other clear lack in the ETSI model is that it does not take in to the consideration the direction of the packets. Since the measured WWW traffic has great unsymmetry and also the used protocols can differ between Uplink and Downlink. With WWW the Uplink the procedure would be quite systematic since there is usually only one data packet (the request) per a WWW-item and the rest are fixed size maintenance packets, whose amount depends about the used protocol and the amount of downloaded data packets. Today many WWW-pages are often composed of several (on average 4.5) WWW-items, so the amount of the items per page could also be used as one variable.

The time interval between two consecutive packets inside a packet call, D_d: This is a geometrically distributed random variable with a mean *?$_{Dd}$* [model time steps], i.e.,

$$D_d \in Geom(\mu_{Dd}) .$$

Naturally, if there are only one packet in a packet call, this is not needed.

Packet size, S_d: The traffic model can use such packet size distribution that suits best for the traffic case under study. Pareto distribution is used.

The Pareto distribution is defined by:

$$\begin{cases} f_x(x) = \dfrac{\alpha \cdot k^{\alpha}}{x^{\alpha+1}}, x \geq k \\ F_x(x) = 1 - \left(\dfrac{k}{x}\right)^{\alpha}, x \geq k \\ \mu = \dfrac{k\alpha}{\alpha - 1}, \alpha > 1 \\ \sigma^2 = \dfrac{k^2 \cdot \alpha}{(\alpha-2)\cdot(\alpha-1)^2}, \alpha > 2 \end{cases}$$

Table 1.1 gives default mean values for the distributions of typical www service. According to the values for α and k in the Pareto distribution, the average packet size µ is 896 bytes. Average requested filesize is $?_{Nd}$ x µ = 15 x 896 bytes ≈ 13,4 kBytes. The interarrival time is adjusted in order to get different average bit rates at the source level. The packet size is limited by a maximum value of 1 Mbyte divided by the average number of packets, i.e 1 Mbyte / 15 ≈ 67 kbyte, giving a finite variance to the distribution.

Table 1.1 Characteristics of connection-less information types

Packet based information types	Average number of packet calls within a session	Average reading time between packet calls [s]	Average amount of packets within a packet call []	Average interarrival time between packets [s][1]	Parameters for packet size distribution
WWW surfing	5	12	15	0.96 0.24 0.12 0.05 0.02 0.004	k = 81.5 α = 1.1

[ref 1] Anderlind Erik and Jens Zander " A Traffic Model for Non-Real-Time Data Users in a Wireless Radio Network" IEEE Communications letters. Vol 1 No. 2 March 1997.

[ref 2] Miltiades E et al. “A multiuser descriptive traffic source model” IEEE Transactions on communications, vol 44 no 10, October 1996.

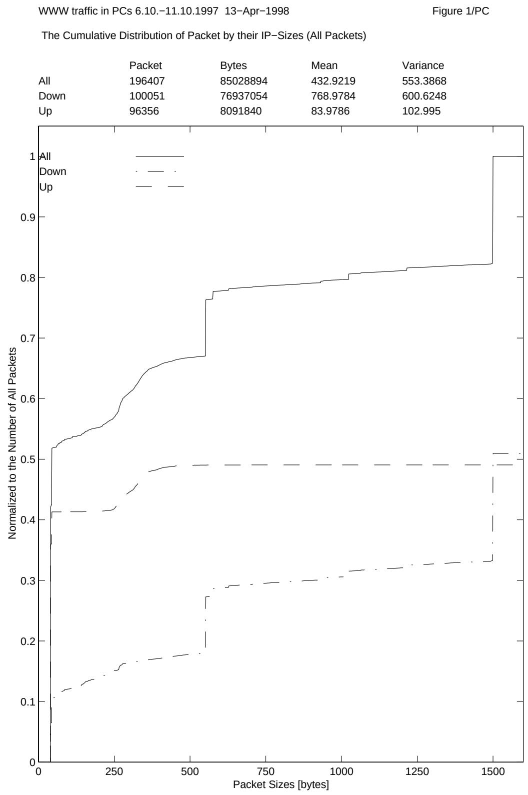
[1] The different interarrival times correspond to average bit rates of 8, 32, 64, 144, 384 and 2048 kbit/s.

4 LAN WWW-TRAFFIC MEASUREMENT DATA SUITED TO ETSI MODEL

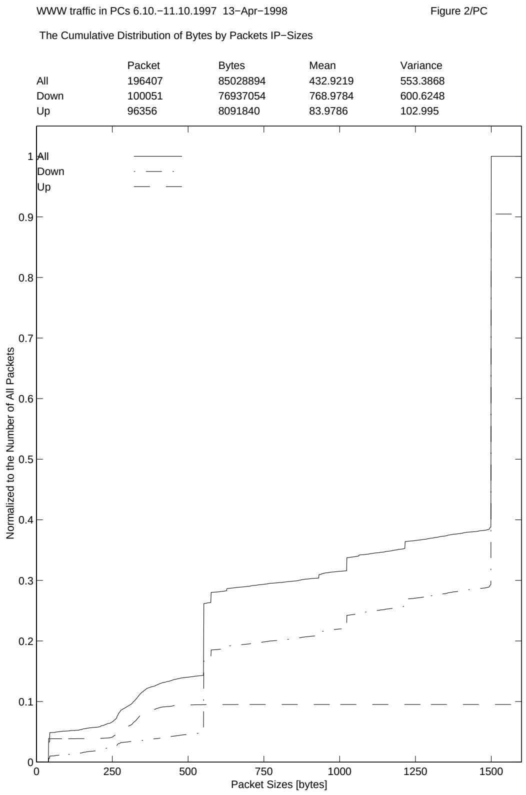
The following model will be based to the reference measurements done 6. - 11. 10. 1997 for a group of PC-machines in Communications Laboratory of HUT. They don’t have so large statistical material than measurements done in 1.8. -26.9.1996, but they reflect better the current situations in this world of changes. Especially maximum packet size 1500 instead of 1024 in -96 and the effect of keep-alive TCP-connections with HTTP 1.1 –protocol make them more interesting. For comparison and time perspective the 1996 results have been added in parenthesis (), when available.

4.1 Packet size, S_d

The documented material includes two figures. In “The Cumulative Distribution of Packets by their IP-Sizes” (fig. 1) all IP-packets are analyzed. “The Cumulative Distribution of Bytes by their IP-Sizes” (fig. 2) shows, how the amount of bytes transferred by these IP-packets is divided to different packet sizes.



	Packet	Bytes	Mean	Variance
All	196407	85028894	432.9219	553.3868
Down	100051	76937054	768.9784	600.6248
Up	96356	8091840	83.9786	102.995



	Packet	Bytes	Mean	Variance
All	196407	85028894	432.9219	553.3868
Down	100051	76937054	768.9784	600.6248
Up	96356	8091840	83.9786	102.995

Every packet includes IP- and TCP-headers (usually 20+20=40 bytes), and about half of them has also a data part. About 80 % of packets do belong to just three fixed size categories defined by the protocol parameters. So there is hardly any “nice” analytical distribution, that would fit very well to them. Best simple model would be in five parts as follows.

1. 51.50 %, size 40/44 bytes (41.31 % up 10.19 % down)
2. 17.63 %, size 1500 bytes (all down)
3. 10.55 %, size 552/576 bytes (all down)
4. 12.57 %, uniform distribution, size 41-1499 bytes, (all down)
5. 7.75 %, normal distribution, mean 275, variance 60 (all up)

An other method would be just to model the downloaded file sizes (WWW-pages or WWW-items) by analytical distribution models. Packet call size would then be the basic random variable and packet sizes and directions would be derived from that through the normal TCP/IP-structure (model for WWW-item).

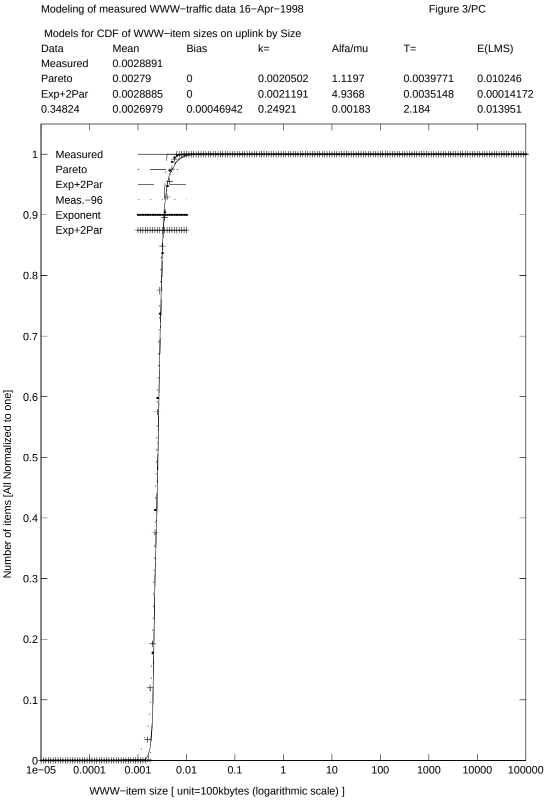
1. Send Request. Size is normally distributed with mean 276 and variance 60 (up)
2. Repeated request (only 5 % of items, up)

   Evaluating the response file size S_{id} and the maximum packet data size P_{max} used by TCP/IP. For non-zero WWW-items the mean is 5671 and the variance is 29586 bytes. A model distribution is presented in next chapter. P_{max} can be set to 512 bytes (IP 552) for 20 % and 1460 bytes (IP 1500) for 80 % of WWW-items. Save the number of maximum size response packets P_{id} = int(S_{id} / P_{max}) to a counter.
3. If the counter is zero go to step 5. Else send a packet size P_{max} (down) and decrement the counter. With probability 48 % repeat this step and with 52 % go to next.
4. Send ACK 40 bytes (up). Repeat this step with probability of 8 % and go to step 3 with probability of 92 %.
5. Send last packet of response. It's size is $S_{id} - P_{id} * P_{max}$ (down).

The distribution of WWW-item Uplink size S_{iu} and Downlink size S_{id} is presented in fig. 3 and 4. The first is quite simple and I believe a normal distribution like step 1 above would do for S_{iu}. Just few repetitions make the distribution a little Pareto like. S_{id} is quite complicated and does not fit well to any single analytical distribution. A reasonable good fitting was reached with a four-part model, which included

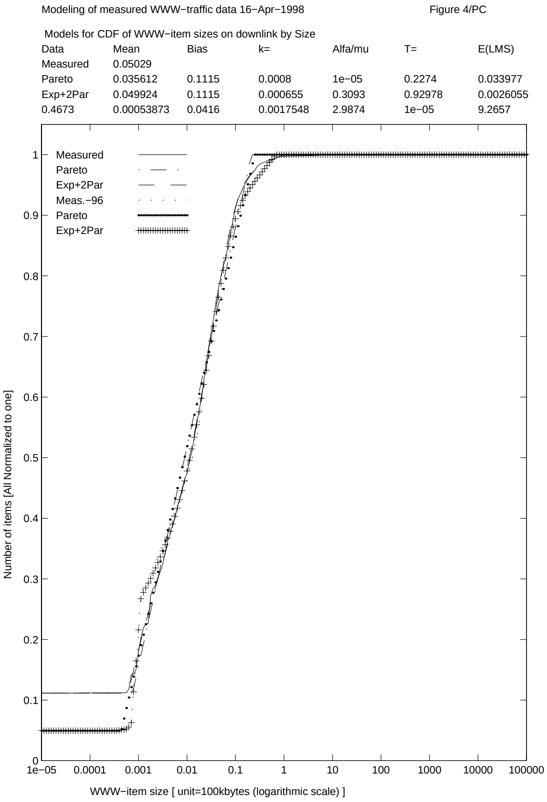
1. 11.15 % of zero sized items (requests that are not responded)
2. 46.73 % exponentially distributed with mean 4160 bytes, when the distribution is shifted to start from 54 bytes
3. 41.94 % Pareto distributed with k=76, alpha=0,3093 and T=92 978
4. 0.18 % Pareto distributed with k=298 740, alpha=0,00001 and T= 926 570

This mixture gives a visually quite nice fit to the measured cdf. It also gets extremely small value 0.0026 (~13 ppm of “power”) for the sum of squared error counted in 201 point geometrically spaced 20 points per decade in the area 1 to 10^{10} . So the error in fitting corresponds fixed mistake of 0.36 % the full amplitude of over the whole area.


Modeling of measured WWW−traffic data 16−Apr−1998 Figure 3/PC

Models for CDF of WWW−item sizes on uplink by Size

Data	Mean	Bias	k=	Alfa/mu	T=	E(LMS)
Measured	0.0028891					
Pareto	0.00279	0	0.0020502	1.1197	0.0039771	0.010246
Exp+2Par	0.0028885	0	0.0021191	4.9368	0.0035148	0.00014172
0.34824	0.0026979	0.00046942	0.24921	0.00183	2.184	0.013951


Modeling of measured WWW−traffic data 16−Apr−1998 Figure 4/PC

Models for CDF of WWW−item sizes on downlink by Size

Data	Mean	Bias	k=	Alfa/mu	T=	E(LMS)
Measured	0.05029					
Pareto	0.035612	0.1115	0.0008	1e−05	0.2274	0.033977
Exp+2Par	0.049924	0.1115	0.000655	0.3093	0.92978	0.0026055
0.4673	0.00053873	0.0416	0.0017548	2.9874	1e−05	9.2657

Also this model’s estimates for mean 4.99 kB and variance 26.95 kB are very close to the mean 5.03 kB and variance 27.75 kB estimated from the measured data.

The final adjustment of the last two parameters could be done “manually” by fitting the area of over 100 Kbytes item size with the measured results. This fourth part included only 0.18 % of items, but caused an increase of 25 % with the mean and 200% with the variance.

For comparison a similar model was fitted to the PC-measurements from summer 1996, and there the model became

1. 4.95 % of zero sized items (requests that are not responded)
2. 30 % exponentially distributed with mean 850 bytes, when the distribution is shifted to start from 70 bytes
3. 43.05 % Pareto distributed with k=8020, alpha=0.8032 and T=230 990
4. 22 % Pareto distributed with k=80, alpha=0.0175 and T=130

The sum of squared error was 0.0059 and the models estimate for mean was 6.02 KB and for the variance 18.64-kB and the mean and variance estimated from the measured data was 5,98 KB and 47,4 KB.

When the whole cdf was tried to model without removing the 11 % bias, the sum of squared error stayed around 0.4 … 1.1. When the bias was removed the situation improved. The major difficulty in fitting the measured data distribution to analytical model is caused by the wide dynamic area of the measured distribution. This makes it difficult to create a model that would simultaneously fit to all 3 ...7 areas found with different slope in the cdf curve calculated from the measured data.

Exponential distribution succeeded poorer with the squared error 0.2827 (0.6809), and the mean 2,18 (1,98) kB and variance 2,44 (2,08) kB were just a fraction of the measured ones.

The Pareto distribution got the squared error of 0.0340 (0.0771) and the mean 3.57 (4.07) kB was about 70 % but the variance 5.30 (6,29) kB only less than 20 % of the measured ones. The values were k=30, alpha=0.00001 and T=22 740 (27 510).

The exponential distribution has a rather fixed form, that achieves the best result when it adjusts the steepest slope in the middle to the over all shape of distribution. The models tend to have much too low values for the mean and variance, since they are defined practically by the small group of large items forming a small tail. This is better modeled with Pareto distribution. Pareto again assumes a linearly decreasing behavior in the log-log-scale that is usually true only to the large items. I used a truncated Pareto distribution that has three parameters and a shifted exponential distribution, which has only two. Pareto seems to be more flexible to adapt to various shapes. Also the tendency of many distributions to rise fast, when things start to happen favors the usage of Pareto.

The model above is not a nice analytical function that would be easy to handle in further analysis or simulations. This seems to be mainly a consequence of the idea of trying to model the data measured from real living network. The results include many “non-idealities” like the 11 % (5 %) of items with no data download. These are probably cases were the server gets the request, but does not from various reason send any response before the next request is send. I have identified one such case, when terminal send a new request on a TCP-connection, where the WWW-server had already started to close the connection. Also there seems to be a noticeable 25 % (32.6%) amount of items with size of 60-400 bytes. They include small, frequently used items like buttons but also announcements about cache hits and error messages. Together they form 36 (38) % of all items.

4.2 The time interval between two consecutive packets inside a packet call, D_d

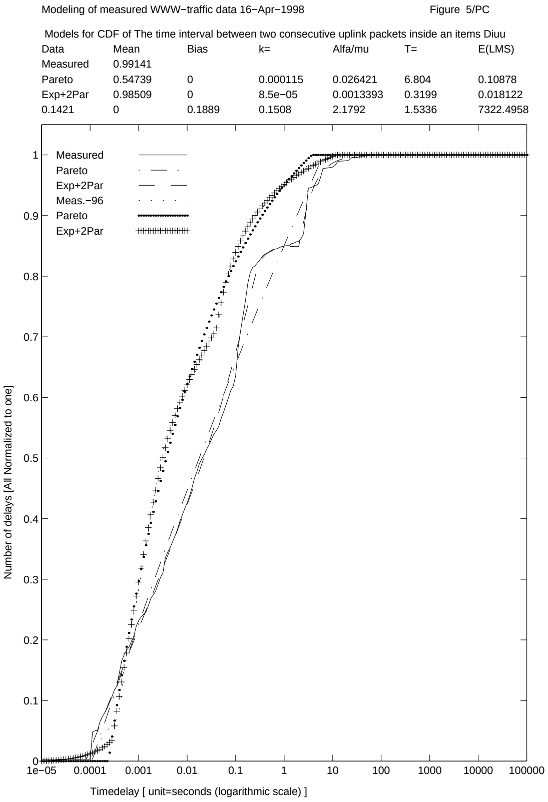
The time interval between two consecutive packets inside a packet call D_d is actually a combination of all the subdistributions caused by the different mechanisms working during a packet call. They create also inherent correlation that affect to the behavior of the traffic. Probably the most important of these is the differences between direction of the transmission. In WWW-traffic the number of packets send from the user to the server is roughly equal to the number of the packet received, but their mean size relates one to ten. Also the timing has great difference, when we compare continuous data flow to receiving an acknowledgement or to recovery after timeout. The behavior of the transmission mechanism is most easy to look inside a WWW-item.

So I analyzed the time distributions inside WWW-items corresponding the four transactions needed to describe sending of an item with two state Markov-process, which is either sending packet to Uplink or receiving a packet from Downlink. The time seems to be clearly smaller after a received packet to get next packet or send a packet, usually acknowledgement. This is mainly caused by the thing that we are observing the traffic so close to the user interface. When we measure time from Uplink packet to Downlink packet, the round trip time of the connection will be added to the server’s response time.

The fifth distribution is the time interval between two consecutive items inside a WWW-page. The structure of WWW-items inside a WWW-page is a tree type of structure a little like directory system in a hard disk of a PC. The whole page can form one item or it can have several extra items like pictures. With new WWW-tools there are already marks about third layer coming to the tree. The WWW-page will start with the “root” item. Then the downloaded data may bring instructions to get other WWW-items. The number of WWW-items per WWW-page can be estimated using $\boldsymbol{N_i}$ and the delay by using $\boldsymbol{D_{pii}}$.

When a model is used to analyze protocol behavior like MAC or RLP the main emphases should be put to the capability of describing the bulk of cases with reasonable accuracy. Especially the short delays, which really are loading the protocol, should not be neglected, like models based on means often do. A small portion of very long delays, which in many cases can be caused by errors or other anomalies, can have a major affect to the means and variances. So it would be more proper to include them somehow to the idle periods.

4.2.1 The time interval between two consecutive Uplink packets inside an item D_{iuu}


Modeling of measured WWW−traffic data 16−Apr−1998 Figure 5/PC

Models for CDF of The time interval between two consecutive uplink packets inside an items Diuu

Data	Mean	Bias	k=	Alfa/mu	T=	E(LMS)
Measured	0.99141					
Pareto	0.54739	0	0.000115	0.026421	6.804	0.10878
Exp+2Par	0.98509	0	8.5e−05	0.0013393	0.3199	0.018122
0.1421	0	0.1889	0.1508	2.1792	1.5336	7322.4958

Here are the models for the time interval between two consecutive Uplink packets inside an item $\boldsymbol{D_{iuu}}$ (fig. 5).

A simple one would just be truncated Pareto distribution with parameters with k=0.000115, alpha=0.0264 and T=6.804 (k=0.0003, alpha=0.2102 and T=4.1286).

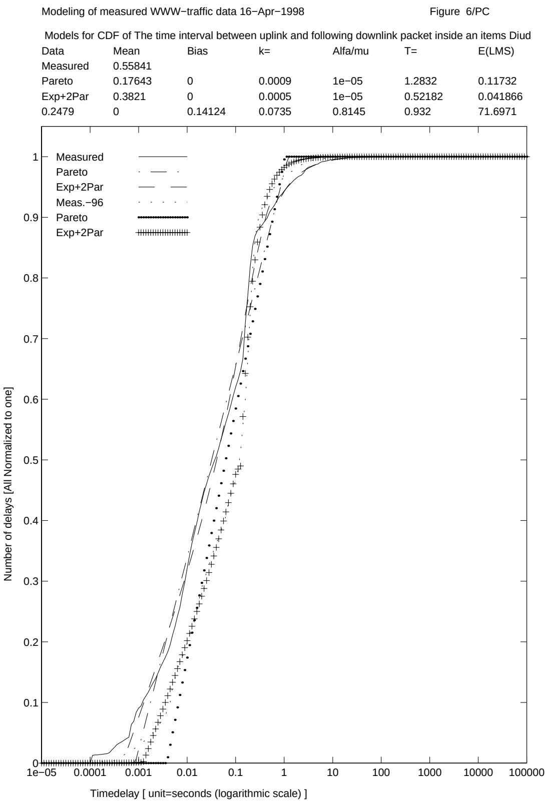
In logarithmic time scale it forms a line (curve), that finds an average between knees of cdf and fits pretty well to the first 68 (65) % of delays. But the mean 0.5484 (0.1652) and variance 1.2434 (0.5232) seconds are clearly smaller than the measured 0.9914 (0.5390) and 4.0852 (28.2766) seconds. The sum of squared error is 0.1088 (0.0402).

A more accurate model would be following

1. 14.21 (15.21) % exponentially distributed with mean 0.1889 (0.0015) s
2. 70.71 (74,70)% Pareto distributed with k=0.000085, alpha=0.00134 and T=0.3199 (k=0.0003, alpha=0.2412 and T=11.0649)
3. 15.08 (10.09)% Pareto distributed with k=2.1792, alpha=1.5334 and T=7322.5 (k=0.0435, alpha=1.4049 and T=210.83 e+6)

The sum of squared error was 0.0181 (0.0090) and the models estimate for mean was 0.9851 (0.2369) s and for the variance 3.5953 (2.2677) s.

4.2.2 The time interval between Uplink and following Downlink packet inside an item D_{iud}



Modeling of measured WWW−traffic data 16−Apr−1998 Figure 6/PC

Models for CDF of The time interval between uplink and following downlink packet inside an items Diud

Data	Mean	Bias	k=	Alfa/mu	T=	E(LMS)
Measured	0.55841					
Pareto	0.17643	0	0.0009	1e−05	1.2832	0.11732
Exp+2Par	0.3821	0	0.0005	1e−05	0.52182	0.041866
0.2479	0	0.14124	0.0735	0.8145	0.932	71.6971

Here are the models for the time interval between Uplink and following Downlink packet inside an item ***D_{iud}***. (fig 6).

A simple one would just be truncated Pareto distribution with parameters with k=0.0009 (0.0040), alpha=0.00001 and T=1.2832 (1.0878).

In logarithmic time scale it forms a line, that fits reasonable well the area 8 … 97 (10 …98) % of delays, but the mean 0.1761 (0.1934) and variance 0.2867 (0.2612) seconds are clearly smaller than the measured 0.5584 (0.4362) and 11.5621 (27.5556) seconds. The sum of squared error is 0.1173 (0.2810)

A more accurate model would be following

1. 24.79 (11.58) % exponentially distributed with mean 0.14124 (0.1193) s
2. 67.86 (41,16)% Pareto distributed with k=0.0005 (0.0014), alpha=0.00001 and T=0.5218 (0.1103)
3. 7.35 (47.26)% Pareto distributed with k=0.8145, alpha=0.9320 and T=71.6971 (k=0.1341, alpha=1.5902 and T=304.93 e+6)

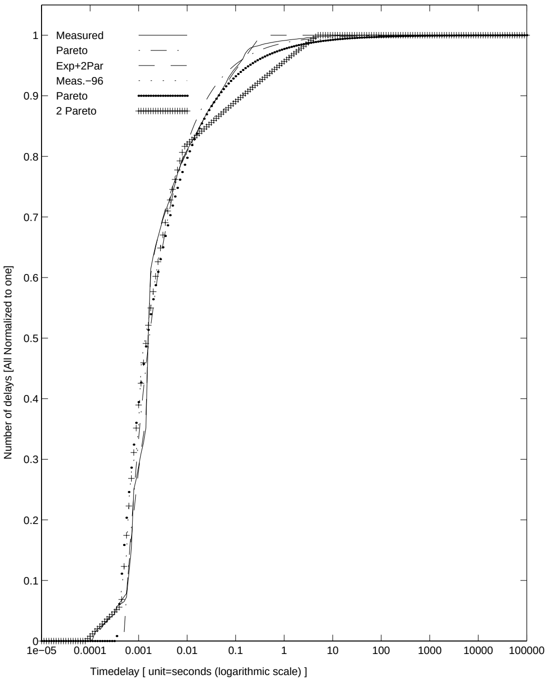
The sum of squared error was 0.0419 (0.0197) and the models estimate for mean was 0.3821 (0.1946) s and for the variance 2.2336 (3.2417) s.

4.2.3 The time interval between two consecutive Downlink packets inside an item D_{idd}

Modeling of measured WWW−traffic data 16−Apr−1998 Figure 7/PC

Models for CDF of The time interval between two consecutive downlink packets inside an items Didd

Data	Mean	Bias	k=	Alfa/mu	T=	E(LMS)
Measured	0.083426					
Pareto	0.26122	0	0.00052019	0.56957	514.7474	0.089935
Exp+2Par	0.018585	0	0.0006	0.9471	0.092189	0.0059881
0.1987	0.0016	8.5e−05	0.3834	0.000115	1e−05	0.3538



Here are the models for the time interval between two consecutive Downlink packets inside an item ***D_{idd}*** (fig. 7).

A simple one would just be truncated Pareto distribution with parameters with k=0.00052, alpha=0. 56957 and T=514. 7474 (k=0.0004, alpha=0. 4761 and T=26300).

In logarithmic time scale it forms a curve, that fits pretty well to the area 6…100 (6…85 and 97…100) % of delays, but the mean 0.2612 (4.3682) and variance 6.4792 (198.58) seconds are clearly larger than the measured 0.0834 (0. 3457) and 1.6570 (12.2385) seconds. The sum of squared error is 0.0899 (0.0710)

A more accurate model would be following

1. 19.87 (0) % exponentially distributed with mean 0.000085 s, when the distribution is shifted to start from 0.0016 s
2. 41.79 (65,91) % Pareto distributed with k=0.0006, alpha=0. 9471 and T=0. 09219 (k=0.0005, alpha=0. 5633 and T=0.0090)
3. 38.34 (34.09) % Pareto distributed with k=0.000115, alpha=0.00001 (0.0264) and T=0.3538 (5.2313)

The sum of squared error was 0.0060 (0.0086) and the models estimate for mean was 0.0186 (0.1444) s and for the variance 0.0517 (0.5901) s.

4.2.4 The time interval between Downlink and following Uplink packet inside an item D_{idu}

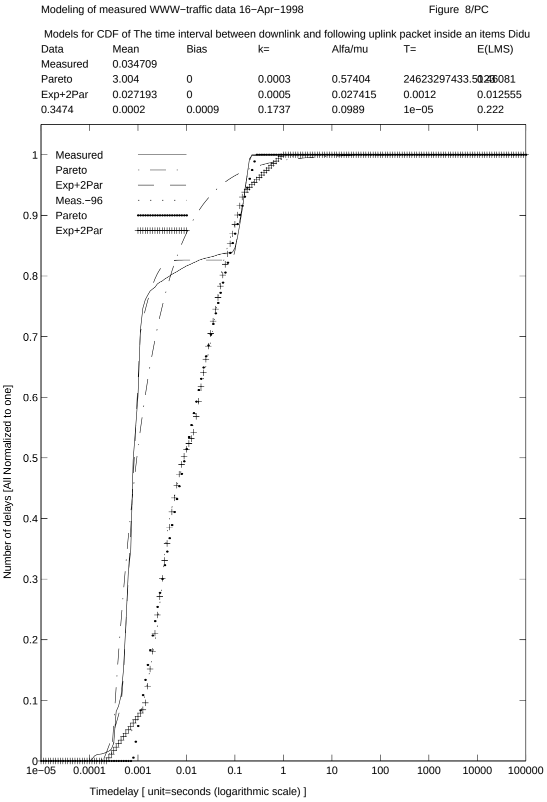
Here are the models for the time interval between Downlink and following Uplink packet inside an item ***D_{idu}*** (fig. 8). In the mew measurements the delays seem to be clearly shorter only from 5 to 50 % of the old ones in the area of 10 … 80 % of cdf. This may be a consequence from the updating of operating system and TCP/IP-stack.

A simple model would just be truncated Pareto distribution with parameters with k=0.0003 (0.0008), alpha=0. 5467 (0.1058) and T=24.623 e+9 (0.2896). In logarithmic time scale it forms a curve that fits to area 2 … 60 % of delays and then tries to round a 17 % step around 100 ms (line, that fits pretty well to the area 5 … 96 % of delays). The mean 3.004 (0. 0395) and variance 539.38 (0. 0622) seconds are clearly larger (smaller) than the measured 0.0347 (0.1164) and 1.0539 (14.01) seconds. The sum of squared error is 0. 4608 (0. 0413)

A more accurate model would be following

1. 34.74 (34.76) % exponentially distributed with mean 0. 0009 (0. 0027) s, when the distribution is shifted to start from 0.0002 (0.0015) s
2. 47.89 (31,03) % Pareto distributed with k=0.0005 (0.0148), alpha=0.027415 (0.3400) and T=0. 0012 (0.1706)
3. 17.37 (34.21) % Pareto distributed with k=0.0989, alpha=0.00001 and T=0.2220 (k=0. 0002, alpha=0.0722 and T=1.0963)

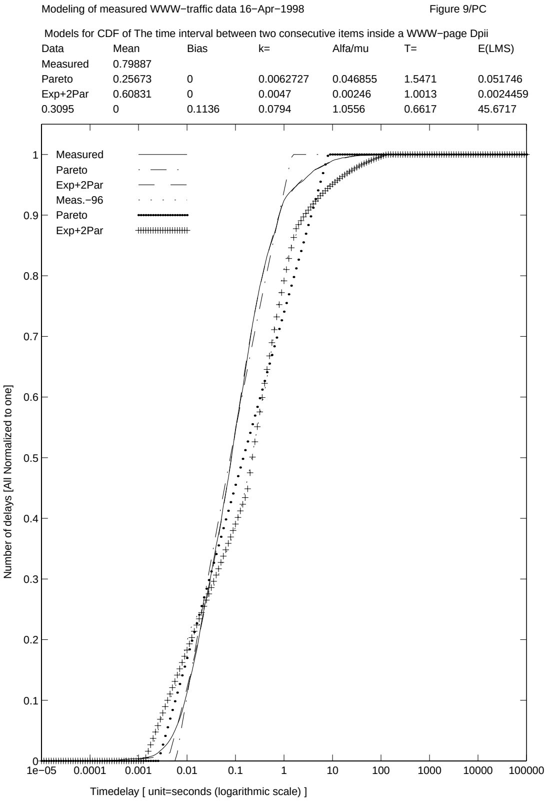
The sum of squared error was 0.0126 (0.0062) and the models estimate for mean was 0. 0272 (0. 0532) s and for the variance 0.0592 (0.1305) s.



Modeling of measured WWW−traffic data 16−Apr−1998 Figure 8/PC

Models for CDF of The time interval between downlink and following uplink packet inside an items Didu

Data	Mean	Bias	k=	Alfa/mu	T=	E(LMS)
Measured	0.034709					
Pareto	3.004	0	0.0003	0.57404	24623297433.5	0.46081
Exp+2Par	0.027193	0	0.0005	0.027415	0.0012	0.012555
0.3474	0.0002	0.0009	0.1737	0.0989	1e−05	0.222



Modeling of measured WWW−traffic data 16−Apr−1998 Figure 9/PC

Models for CDF of The time interval between two consecutive items inside a WWW−page Dpii

Data	Mean	Bias	k=	Alfa/mu	T=	E(LMS)
Measured	0.79887					
Pareto	0.25673	0	0.0062727	0.046855	1.5471	0.051746
Exp+2Par	0.60831	0	0.0047	0.00246	1.0013	0.0024459
0.3095	0	0.1136	0.0794	1.0556	0.6617	45.6717

4.2.5 The time interval between two consecutive items inside a WWW-page D_{pii}

Here are the models for the time interval between Downlink and following Uplink packet inside an item D_{pii} (fig. 9). A simple one would just be truncated Pareto distribution with parameters with k=0.0063 (0.0027), alpha=0.0469 (0.00001) and T=1.5471 (8.5698). In logarithmic time scale it forms a line that fits well to area 6 …

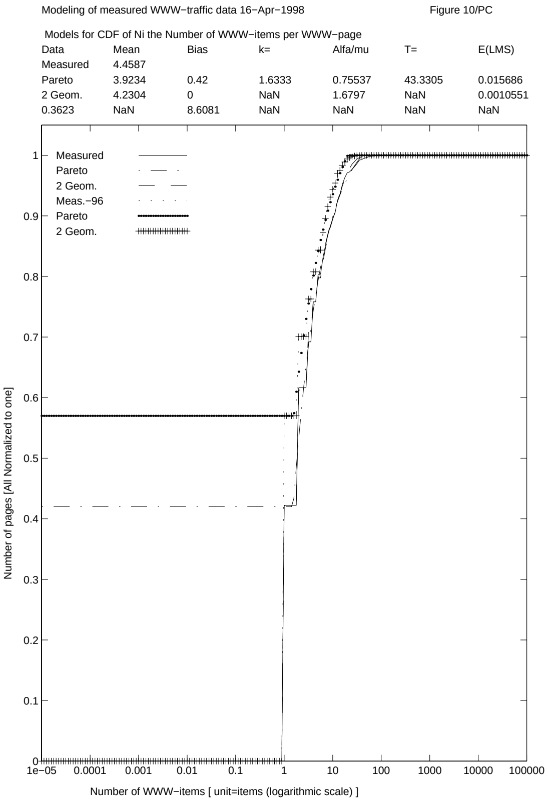
92 % of delays. The mean 0.2567 (1.0627) s and variance 0.3593 (1.8500) s are clearly larger (smaller) than the measured ones 0.7989 (7.2475) s and 10.3926 (430.4064) s. The sum of squared error is 0.0517 (0.1541)

A more accurate model would be following

1. 30.95 (4.06) % exponentially distributed with mean 0.1136 (0.5162) s
2. 61.11 (62.71) % Pareto distributed with k=0.0047 (0.0015), alpha=0.00246 (0.0152) and T=1.0013 (1.8393)
3. 7.94 (33.23) % Pareto distributed with k=1.0556 (0.1839), alpha=0.6617 (0.3763) and T=45.6717 (140.2821)

The sum of squared error was 0.0024 (0.0067) and the models estimate for mean was 0.6083 (2.6561) s and for the variance 2.6510 (11.3622) s.

4.3 The number of packets in a packet call, N_d



Modeling of measured WWW−traffic data 16−Apr−1998 Figure 10/PC

Models for CDF of Ni the Number of WWW−items per WWW−page

Data	Mean	Bias	k=	Alfa/mu	T=	E(LMS)
Measured	4.4587					
Pareto	3.9234	0.42	1.6333	0.75537	43.3305	0.015686
2 Geom.	4.2304	0	NaN	1.6797	NaN	0.0010551
0.3623	NaN	8.6081	NaN	NaN	NaN	NaN

This parameter follows actually as a combination from two random variables size of an item S_i and the number of items per WWW-page N_i. This is based from the assumption, that the downloading of one WWW-page would in practice cause a packet call. The behavior and division into packets of S_i is described in chapter 4.1.

The distribution for N_i (fig. 10) is rather sort and simple and with step function of .42 (.57) in n=1. In fig 10 and 11 this forms horizontal lines from 1e-5 to 1 that should be removed. A Pareto distribution would model it also without step function reasonably well with k= 0.7577 alpha=0.7399 and T=50.51 (k=0.7443, alpha=1.0495 and T=187.78). The sum of squared error was 0. 1075 (0.2041) and the model estimates were for mean 4.4665 (3.7863) and for the variance 7. 0598 (10.1213), when the mean and the variance estimated from the measured data were 4.4587 (3.1180) and 7.6006 (4.6329).

Almost a half (42,2% -97 and 57 % -96 in measurements) of WWW-pages include only a single WWW-item. This very sharp beginning was the primary difficulty in fitting the measured data to Pareto distribution. So separating this single item to a separate delta function to value one, would be the main improvement in improving the accuracy of fitting.

1. 42 % (57%) of single items
2. 57.78 % (43 %) Pareto distributed with k=1.6333, alpha=0.7576 and T=43.6889 (k=1.6582, alpha=0. 57 and T=21.4159)

With this enhancement the sum of squared error would reduce to 0.0157 (0.0076) and the corresponding estimates are 3.9285 (2.4656) for the mean and 6.6749 (4.1328) and for the variance.

Since ETSI model uses geometric distribution to model the number of packets in a packet call, I tried it also that approach. Single geometric distribution has the best fit in least mean square sense, when 1/p=2.8368 (2.0871). The mean 2.8742 (2.3156) and especially the variance 2.3156 (1.5371) are clearly too small and the sum of square error is 0.1289 (0.1641). When the mean is set to the nominal, the variance stays still small 3.9693 (1.5371) and the sum of square error rises to 0.5304 (0.4990).

Dividing the distribution into sum of two geometric distributions gives very good fitting. Partly this is consequence from the fact that time discrete distribution is synchronous with the measured data unlike continuous distributions like Pareto. The sum of square error for the two stage model optimized by fitness is 0.0010551 (0.0002) and the components are

1. 63.77 (56.39) % geometrically distributed with 1/p=1.6797 (1.1525)
2. 36.23 (43.61) % geometrically distributed with 1/p=8.6081 (5.3996)

The mean is 4.2304 (3.0329) is in side ±5 % tolerance and variance is 6.0011 (3.8993). For the number of items it is important that the model reaches at least roughly the mean. Otherwise the load caused by using the model would be less than the actual measured. So I would recommend either the simple Pareto model although it is a little less accurate in the details or double geometrical.

4.4 The reading time between two consecutive packet call requests in a session, D_{pc}

The reading time between two consecutive packet call requests (WWW-pages) in a session ***Dpc*** (fig.11) is a human related factor. In principle it depends only about user behavior, although the way in which the information is divided to WWW-pages will also affect it. Human behavior can be assumed to be more stable than technical. Still it is almost surprising how same the mean and variance are in measurements from 1996 and 1997.

A simple model for the reading time between two consecutive packet call requests (WWW-pages) in a session D_{pc} would be a truncated Pareto distribution with parameters with k=1.1042 (1.3490), alpha=0.1363 (0.0005) and T=145.7542 (98.3536). In logarithmic time scale it forms a curve, that fits very well to cdf (a line that fits pretty well to the shortest 92 % of delays). The mean 23.9436 (22.5877) seconds is just 9 (14) % and the variance 32.6850 (24.8605) seconds 24 (37) % smaller than the measured ones 26.2870 (26.3295) s and 43.0555 (40.0927) seconds. The sum of squared error is 0.0093 (0.0272) is also reasonable good.

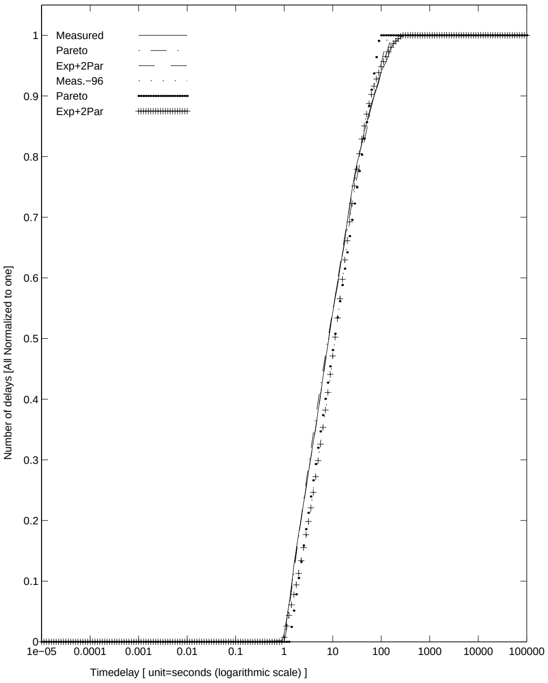
A more accurate model would be following

1. 1.76 (43.95) % exponentially distributed with mean 8.2279 (18.7389) s, when the distribution is shifted to start from 0 (1.9752) s
2. 89.35 (46,05)% Pareto distributed with k=1.0490 (1.0146), alpha=0.1649 (0.2873) and T=177.1651 (279.3458)
3. 8.89 (10.00)% Pareto distributed with k=5.1192 (3.7166), alpha=0.00001 and T=34.9593 (179.6776)

Modeling of measured WWW−traffic data 16−Apr−1998 Figure 11/PC

Models for CDF of The time interval between WWW−pages inside a WWW−session Dpc

Data	Mean	Bias	k=	Alfa/mu	T=	E(LMS)
Measured	26.287					
Pareto	23.9436	0	1.1042	0.1363	145.7542	0.0093209
Exp+2Par	24.6897	0	1.049	0.1649	177.1651	0.0015448
0.0176	0	8.2279	0.0889	5.1192	1e−05	34.9593



The sum of squared error was excellent 0.0015 (0.0004) and the models estimate for mean was 24.6904 (26.2728) and for the variance 36.2916 (39.5950) seconds, which are just 6 (0.2) % and 16 (1.2) % smaller than the measured ones. The measured behavior can be assumed stable, the statistics and curves are quite alike and 1996 measurement have much larger amount of samples and better fitting with accurate modeling. These reasons make it a considerable issue, when selecting models to simulate.

4.5 The number of packet call requests per session, ***N_{pc}***

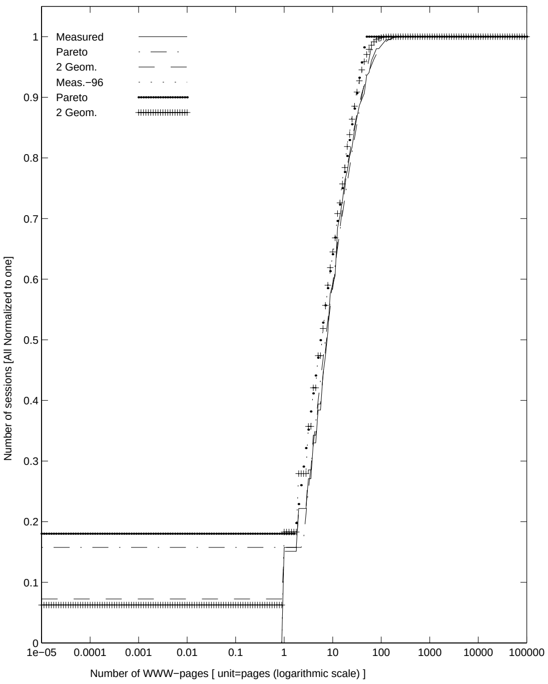
This parameter of the developers of ETSI model I would like connect to the number of WWW-pages in the session. In a ideal case this would be very natural result since without error, congestion etc. the packet flow should be almost continuous during downloading the page, when compared to the interrupt of several seconds the user needs for reading. In practice the errors can interrupt downloading for seconds or minutes and on the other hand a user can in quick scanning click next page already before all items of the present one are downloaded.

The number of pages per WWW-session ***Npc*** is based from the assumption, that the downloading of one WWW-page would in practice cause a packet call. The distribution for ***Npc*** (fig. 12) is rather short and simple like ***N_i*** in chapter 4.3.

Modeling of measured WWW−traffic data 16−Apr−1998 Figure 12/PC

Models for CDF of Npc the Number of WWW−pages per WWW−session

Data	Mean	Bias	k=	Alfa/mu	T=	E(LMS)
Measured	16.2301					
Pareto	14.7204	0.1576	2.56	0.22883	73.8183	0.025224
2 Geom.	16.4001	0.0725	NaN	38.0396	NaN	0.0040381
0.65609	NaN	9.1707	NaN	NaN	NaN	NaN



With a Pareto distribution it can be modeled reasonably well with k=0.8279 (0.7312), alpha=0.0001 and T=66.69 (49.2884). The sum of squared error was 0.0701 (0.0355) and the model estimates were for mean 15.0537 (11.5234) and for the variance 16.7437 (12.4809), when the mean and the variance estimated from the measured data were 16.2301 (13.0138) and 23.3771 (22.3529).

A noticeable part (15.76 % -97 and 18.02 % -96) of WWW-sessions includes only a single WWW-page. This very sharp beginning was the primary difficulty in fitting the measured data to Pareto distribution. So separating this single item to a separate delta function to value one, would be the main improvement, if the accuracy of fitting would be improved.

1. 15.76 (18.02) % of single pages
2. 29.14 (46.51) % % exponentially distributed with mean 7.1039 (13.9632), when the distribution is shifted to start from 1.6699 (1.6766)
3. 30.73 (34.64) % Pareto distributed with k=1.7810, alpha=0.0057 and T=80.2358 (k=1.6603, alpha=0.4871 and T=129.8544)
4. 24.37 (0.83) % Pareto distributed with k=5.8812, alpha=0.6224 and T=188.6 (k=6.5377, alpha=17.7039 and T= 32.988)

With this enhancement the sum of squared error would reduce to 0.0053 (0.0064) and the corresponding estimates are 16.0575 (12.5016) for the mean and 23.2841 (17.4139) and for the variance. For the number of packets calls it is important that the model reaches at least roughly the mean. Otherwise the load caused by using the model would be less than the actual measured.

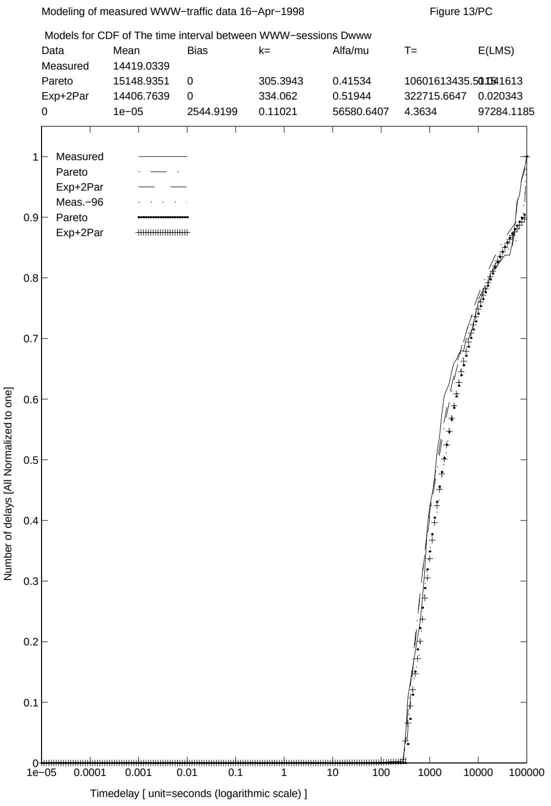
Single geometric distribution has the best fit in least mean square sense, when 1/p=11.7674 (9.2967). The mean 11.8557 (9.3820) and especially the variance 15.7032 (8.8140) are clearly too small and the sum of square error is 0.1111 (0.1444). When the mean is set to the nominal, the variance stays still small 15.7032 (12.5071) and the sum of square error rises to 0.3035 (0.3503).

Dividing the distribution into sum of two geometric distributions gives very good fitting. Partly this is consequence from the fact that time discrete distribution is synchronous with the measured data unlike continuous distributions like Pareto. The sum of square error for the two stage model optimized by fitness is 0.0040 (0.0010) and the components are

1. 7.25 (6.25) % of single pages
2. 65.60 (34.19) % geometrically distributed with 1/p=9.1696 (3.9021)
3. 27.15 (59.56) % geometrically distributed with 1/p=38.0236 (18.0960)

The mean is 16.3987 (12.1761), which is in side +1 (+7) % tolerance and variance 24.7062 (15.5377)

4.6 The passive time between two WWW-sessions, *D_{WWW}*



Modeling of measured WWW−traffic data 16−Apr−1998 Figure 13/PC

Models for CDF of The time interval between WWW−sessions Dwww

Data	Mean	Bias	k=	Alfa/mu	T=	E(LMS)
Measured	14419.0339					
Pareto	15148.9351	0	305.3943	0.41534	10601613435.50	0.041613
Exp+2Par	14406.7639	0	334.062	0.51944	322715.6647	0.020343
0	1e−05	2544.9199	0.11021	56580.6407	4.3634	97284.1185

The time a user or a terminal stays passive between two consecutive WWW-sessions ***Dwww*** (fig.13) depends only about user behavior. Although our samples are quite small in numbers the curves of measurements from 1996 and 1997 are quite alike. This is quite natural since the dynamic area is limited to two and half decades from 300 to 100000 seconds.

A simple model for the passive time between two consecutive WWW-sessions ***Dwww*** would be a Pareto distribution with parameters with k=305.4 (346.3), alpha=0.4153 (0.3664) and T=1.06e+10 (3.09e+06). In logarithmic time scale it forms a curve, that fits pretty well to the first 90 % of delays, and the mean 15149 (16616) and the variance 30048 (30765) seconds are quite close to the measured ones 14419 (16878) and 26248 (31053) seconds.

The sum of squared error 0.0416 (0.0103) is also reasonable.

A more accurate model would be following

1. 0 (3.02) % exponentially distributed with mean 2545 (3136.6) s, when the distribution is shifted to start from 1e-5 (0.0010) s
2. 88.98 (70,62)% Pareto distributed with k=334 (295.9), alpha=0.5194 (0.3842) and T=3.227e+5 (8.9788e+11)
3. 11.02(26.36)% Pareto distributed with k=56580 (643.51), alpha=4.3634 (0.4624) and T=97284 (1.236e+10)

The sum of squared error was good 0.0203 (0.0047) and the models estimate 14407 (16791) for mean and 26297 (31341) for the variance fit very well to the measured ones. Since 1996 measurement has much larger amount of samples and better fitting to models, it could be better choice for simulation.

5 THE WWW-TRAFFIC MEASUREMENTS ON WAVELAN

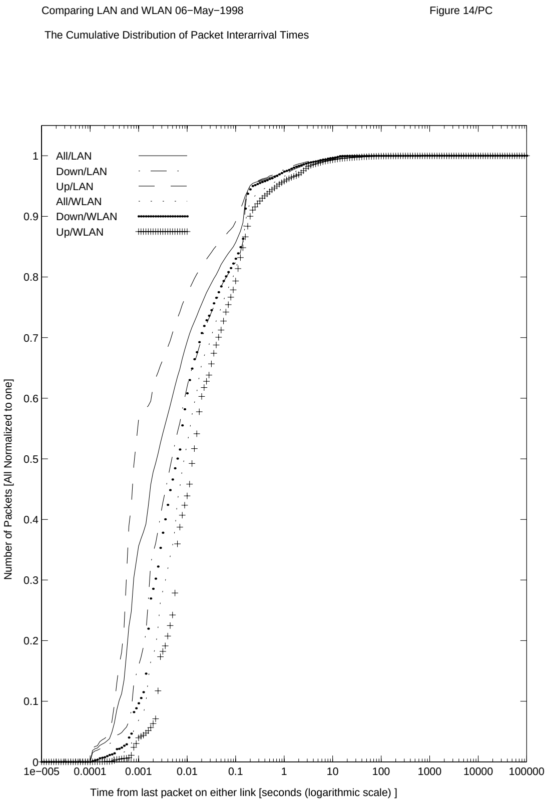
The measurements done on WaveLan were rather short. During the measuring period 6.10. – 11.10.1997 there were two rather extensive 1-2 hour test periods with two portables connected to WaveLan. The aim of these measurements was just try to find the differences caused by extending laboratory's traditional LAN with WaveLan. Simultaneously a reference measurement was done with two PCs connected directly to LAN.

To reduce the error caused by measuring setup, the test persons were using simultaneously two terminals, one portable and one PC and they were instruct to visit the same WWW-pages with both terminals. The pages were looked in small groups and in random order to avoid systematic errors, which would be caused by data left to WWW-servers cache, variations in network loading etc. Test persons switched from one terminal to other after clicking next page to download and they waited until the whole page was downloaded before starting the next one.

Here follows a condensed comparison between the results of portables connected to WaveLan (WLAN) and the reference measurement done for PCs connected directly to LAN (LAN). First is the statistics of both the measurements in table 5.1, which actually is a subgroup of table 2.1.

	LAN	WLAN	%
Packets up	26129	23027	13.47
Packets down	27700	24051	15.17
Data-Packets up	4799	4541	5.68
Data-Packets down	23143	19454	18.96
IP-bytes up [kB]	2308	2263	1.99
IP-bytes down [kB]	21312	17878	19.21
Data-bytes up [kB]	1255	1335	-5.99
Data-bytes down [kB]	20198	16910	19.44
Nibbles	16919	22909	-26.15
Bursts	897	1153	-22.20
WWW-items	4682	4411	6.14
TCP-connections	1764	1676	5.25
WWW-pages	739	780	-5.26
WWW-sessions	9	8	12.50
Nibble time [s]	77.77	91.95	-15.42
Burst time [s]	2617.88	2979.93	-12.15
WWW-item time [s]	7194.38	8113.24	-11.33
TCP-connection time [s]	50157.20	52505.70	-4.47
WWW-page time [s]	3169.54	4086.67	-22.44
WWW-session time [s]	11985.60	13136.1	-8.76

Table 5.1. The main statistics of WaveLan data measured Packets, IP-bytes and Data-bytes, the numbers and total lengths of Bursts, Nibbles, TCP-connections, WWW-items, WWW-pages and WWW-sessions



When we make a comparison of WaveLan measurements to simultaneous reference PC measurements based on table 5.1, it comes obvious that the idea of transfering same WWW-page simultanuously through LAN and WLAN was not totally fulfilled. The number of WWW-items is over 6 % larger in the reference and the number of WWW-pages is over 5 % less. The later can be also caused by the quick response and short mechanical reading times in a measurement, where users are just trying to generate as much traffic as possible. In some cases the user may have clicked the next page already before one second has gone from the previous page was ready.

The amount of packets and bytes is clearly larger in LAN, but the number of nibbles and bursts and the cumulative active time used in all higher structures is clearly less.

This confirms the logical idea that adding a extra network element like WLAN to a connection using end-to-end TCP-protocol, will naturally add delay to the connection. Here are the noticable size differencies found from the basic figures.

In packet sizes WLAN had more (19.17 %) empty packets on on the downlink than LAN (16.49%). The request size on uplink for WLAN is larger. Roughly the mimimum for WLAN in the median for LAN, but maxima are about the same as you can see from the table 5.2. A logical reason for this is that in checking the other PC was found to use Netscape 3.0, when the others used version 4.01.

	min.	10 %	50 %	90 %	max.
Up/LAN	1	213	275	312	673
Up/WLAN	1	272	294	321	428

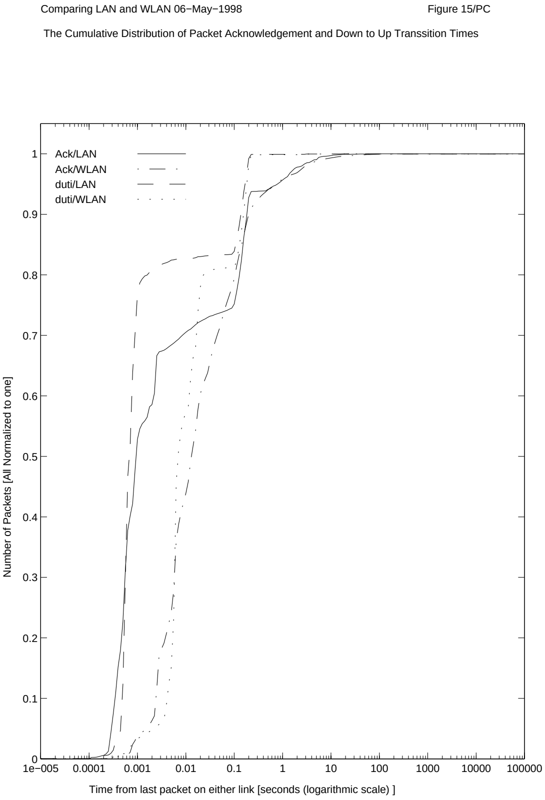
Table 5.2. The packet sizes, where the minimum, the maximum and the limits of 10 %, 50 % and 90 % of the measured Data-Bytes were reached Uplink

Packet interarrival delays are longer with WLAN in uplink when measured from the previous packet on the LAN in either direction. This can be seen most clearly about figure 14 and the table 5.3. At median and actually at the shortest 80% of delays can the values for WLAN are over tenfold compared to LAN, but the means are just 10-15% longer.

This delay is caused by the WavelLan section added between measuring point in LAN and the portable PC terminal. In downlink we can see also some 20-50 % additional delay, although the added WaveLan should not have any direct affect to packets between measuring point in LAN and the WWW-server. Maybe larger response times have some effect to WWW-servers internal priorities.

	min.	10 %	50 %	90 %	Max.
All/LAN	0.000112	0.000398	0.00251	0.158	1e+05
Down	0.000112	0.000794	0.00501	0.158	708
Up	0.000112	0.000355	0.000891	0.112	1e+05
All/WLAN	0.000112	0.00158	0.00891	0.178	1e+05
Down	0.000112	0.00112	0.00631	0.158	708
Up	0.000112	0.00251	0.0141	0.2	1e+05

Table 5.3. The packet interarrival times [s], when the minimum, the maximum and the limits of 10 %, 50 % and 90 % of the measured IP-Packets were reached. The main statistics of WaveLan data measured Packets, IP-bytes and Data-bytes, the numbers and total lengths of Bursts, Nibbles, TCP-connections, WWW-items, WWW-pages and WWW-sessions



n the delays are looked only inside
-connections, the clearest change
at majority of delays under 1 ms
increased to the area from 1 to 15

clear difference can be seen in the
y from a data packet coming in the
nlink to the acknowledgement
ned on the Uplink (Table 5.4 and
15). Since these values are
ulated from the TCP sequence and
owledgement fields, they should
y describe the terminals response
as measured from the local LAN.
e noticed acknowledgements are
ded here. This includes connection
ing, closing and downloading of
W-items. The minimum delay is 0.1
or LAN and 1.8 ms for WaveLan
in until 70 % of delays (7.9 ms
, 100 ms WLAN) the difference
ns to be eight to twenty folds.

When the delays are looked inside WWW-items, which I call just a simple request response pair on a TCP-connection, the differences are also noticeable. Clearest change is that majority of delays under 1 ms have increased to the area from 1 to 15 ms. Very clear difference can be seen in the delay from a data packet coming in the Downlink to the acknowledgement returned on the Uplink (iduti3, Table 5.7 and Fig. 15). This is could form a subgroup to the acknowledgements mentioned above, since every WWW-item uses a single TCP sequence. Although acknowledgements are sent so rapidly, there is always a chance of the next data packet “cutting in” between a Downlink data packet and it’s acknowledgement. With WLAN this possibility is even larger due the additional delay. So the minimum delay is the same about 0. This must be due some acknowledgements from earlier packets that are just waiting in the output buffer of a PC or a WaveLan bridge for LAN to be freed after next data packet. But from 10 % (0.4ms LAN, 4 ms WLAN) to

80 % (1.6 ms LAN, 22.4 WLAN) of cdf the delays of WaveLan seem to be about at least ten times larger than delays in LAN.

	WLAN	LAN	WLAN/LAN
Min	0.0018	0.0001	17.7828
0.1000	0.0025	0.0003	7.9433
0.2000	0.0056	0.0004	12.5893
0.3000	0.0071	0.0005	14.1254
0.4000	0.0126	0.0006	19.9526
0.5000	0.0178	0.0009	19.9526
0.6000	0.0251	0.0020	12.5893
0.7000	0.1000	0.0079	12.5893
0.8000	0.1413	0.1259	1.1220
0.9000	0.1778	0.1778	1.0000
Max	63.0957	100.0000	0.6310

*Table 5.4. Packet acknowledgement from terminal (from Downlink data to Uplink ACK) Times [s], when the minimum, the maximum and the limits of n*10 % of the measured delays were reached and their relation between WLAN and LAN*

Iuuti3	WLAN	LAN	WLAN/LAN
0.0100	0.0007	0.0001	7.0795
0.1000	0.0018	0.0003	7.0795
0.2000	0.0022	0.0020	1.1220
0.3000	0.0050	0.0032	1.5849
0.4000	0.0100	0.0056	1.7783
0.5000	0.0141	0.0089	1.5849
0.6000	0.1000	0.0141	7.0795
0.7000	0.1259	0.0447	2.8184
0.8000	0.3548	0.1259	2.8184
0.9000	1.2589	2.2387	0.5623
1.0000	70.7946	63.0957	1.1220

*Table 5.5. From Uplink to Uplink packet Interarrival times inside WWW-items [s], when the minimum, the maximum and the limits of n*10 % of the measured delays were reached and their relation between WLAN and LAN*

Iudti3	WLAN	LAN	WLAN/LAN
0	0.0001	0.0001	1.0000
0.1000	0.0071	0.0014	5.0119
0.2000	0.0079	0.0040	1.9953
0.3000	0.0112	0.0079	1.4125
0.4000	0.0158	0.0112	1.4125
0.5000	0.0355	0.0200	1.7783
0.6000	0.1259	0.0562	2.2387
0.7000	0.1413	0.1413	1.0000
0.8000	0.1585	0.1585	1.0000
0.9000	0.5012	0.2239	2.2387
1.0000	63.0957	70.7946	0.8913

*Table 5.6. From Uplink to Downlink packet Interarrival times inside WWW-items [s], when the minimum, the maximum and the limits of n*10 % of the measured delays were reached and their relation between WLAN and LAN*

Iduti3	WLAN	LAN	WLAN/LAN
0	0.0000	0.0000	1.0000
0.1000	0.0040	0.0004	8.9125
0.2000	0.0050	0.0005	10.0000
0.3000	0.0056	0.0006	10.0000
0.4000	0.0056	0.0006	10.0000
0.5000	0.0063	0.0006	10.0000
0.6000	0.0112	0.0007	15.8489
0.7000	0.0158	0.0008	19.9526
0.8000	0.0224	0.0016	14.1254
0.9000	0.1413	0.1259	1.1220
1.0000	50.1187	19.9526	2.5119

*Table 5.7. From Downlink to Uplink packet Interarrival times inside WWW-items [s], when the minimum, the maximum and the limits of n*10 % of the measured delays were reached and their relation between WLAN and LAN*

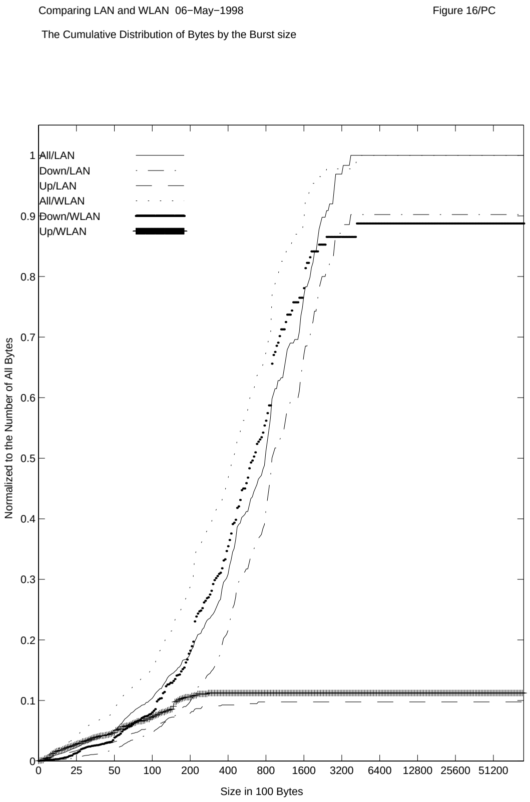
It seems quite obvious that WLAN increases the short (0.1 to 10 ms) delays about with factor of ten, when measured from downlink to uplink. The reasons for that must be looked from the physical setup used in measurements. When a packet comes on the downlink it has almost got through the LAN, when GOBBLER stamps the time. So in the minimum response time of 0.1 ms the LAN transfers only 1000 bits, which is 125 bytes. In Ethernet the minimum MAC packet size is 64 bytes, so about half of the response time goes to putting the packet in to LAN. The difference in response times is probably used in prosessing the received packet. And the processing time seems to be the same class than the time data packet needs going through LAN.

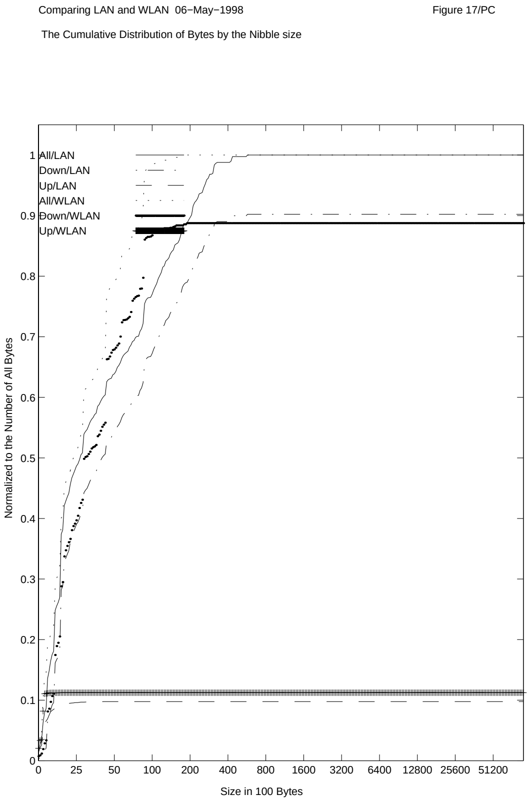
In WLAN it gets only in to the base station (bridge) when stamped in the LAN. So it needs still an other transmission time to reach the portable terminal. Since WaveLans speed is just 2 Mbit/s packets of 64 to 1500 would need 0.256 to 6 ms to go through and also shortes uplink packet needs extra .256 ms. WaveLan uses on the radio path its own prorietary protocol, which can add some delay, and the processing power of portable 486 is less than a desktop Pentium. Collisions and errors in WaveLan will naturally also cause some additional delay. The amount of retransmitted bytes is double for WLAN (LAN 13517 + 35853 = 49370, WLAN 27666 + 69924 = 97590), but they still form less than 1 % of all traffic.

The effects of larger delays can be seen also to accumulate to larger units. In all compatisons LAN takes 5 – 22 % less time to transfer 19 % more data than WLAN.

In bursts and nibbles it’s affects also to the number and size of bursts and nibbles. The burst size was with LAN 60,01 and with WLAN 40.83 packets so burst size seems to be about 33 % smaller. This can be caused by by WLAN’s additional delay, which can cause part of the delays to grow over the fixed 2 second limit. An other thing is that the WWW-items size in WLAN were only 90 % from the LAN. In fig 16. is shown the cumulative distribution of Bytes as a function of Burst size. There can be seen that the gap berween WLAN and LAN opens at Burst sizes from 10 to 20 kB and starts to close above 80 kB.

Similar behavior can be seen at nibble sizes (LAN 3.182 and W-LAN 2.055), where the fixed limiting idle time is just 10 ms. As a function of nibble length the single packets part of downlink byte increases only from 8 % of LAN to 13 % WLAN, while on uplink the change is from 22 to 54 %. Some picture of how the nibbles on the downlink get fragmented can be seem from the distribution of bytes on various nibble sizes (fig17). In both LAN and WLAN about a half of downlink bytes flow in nibbles with size of less than 3 kbytes. But when with WLAN the nibbles from 3 to 9 kbytes carry roughly the same amount of data, in LAN they carry only a quarter and the rest goes with units of 10 to 60 kbytes.





6. CONCLUSION

In chapter 3 I have shown some criticism to the NRT packet data model used in ETSI. The typical features of the used protocols like TCP/IP and HTTP should be noticed, if UMTS is hoped to transfer them efficiently. Especially the datagram size and average interarrival time distributions are in practice so concentrated to certain small areas that analytical models based to measured means can be misleading in some essential parts. An other clear lack is that the directions of the packets so also the great unsymmetry between Uplink and Downlink in WWW traffic and in shorter time scales also in many other Internet services are not taken in consideration.
In chapter 4 I have shown some models at different accuracy, which converge with the measurements I have done. They are mainly intended to simulation. A model with twelve distributions all combined from three or four analytical distributions become very large and hard to implement. In practice only few values needs to be calculated per packet. The ETSI model calculates the size and interarrival time for each packet. In my proposal would be needed the state of next packet (includes the direction), the distribution to be used and the value for interarrival time. And two out of these three could be quite simple probabilities of selecting next state. The measurements will surely give different results in different places and times. But I

don't believe that the networks and protocols used in future would be considerable slower and most studies expect that the file sizes continue to grow. So it would feel safer to base the future solutions on data, where the shorter delays and larger file sizes and item numbers would be rather emphasized than neglected.
In chapter 5 I have reported measurements, where portables connected to WaveLan were compared to PCs directly connected to LAN. As could be expected adding wireless LAN to series with traditional LAN will increase delay and also errors. WaveLAN increases the short 0.1 to 10 ms delays with factor of ten, when measured from Downlink to Uplink. The main reason for this is the physical transmission delay caused by adding an extra link with slower 2 Mbit/s bandwidth. The mean size of bursts decreases with about one-third in two different time scales. The fractionating decreases clearly to the group of largest bursts and increases the middle group, while the relative part of small bursts does not change.

This report has been made to give the measured results, and the necessary background information to the use of other researchers and projects. Due the hurry there are some errors, and the evaluation of the results is still partly based to intuition. The most important result is fitting the measured data with rather good accuracy to a model of known mechanisms of both HTTP and TCP/IP protocols. The major draw back is the amount of parameters and distributions that makes the model rather large and complicated. Much work would be needed to find both the possibilities to simplify it and the effects of simplifications to the accuracy.

7 REFERENCES

[1] Mah, B, An Empirical Model of HTTP Network Traffic, Proceedings of INFOCOM'97, Kobe, Japan, April 7-11 1997.

[2] Nieminen, T., Report on WWW-traffic measurements, Helsinki University of Technology, Communications Laboratory, Technical Report T39, 28. October 1996.

[3] Aldén M., Traffic Models for WWW User-Behaviour from the Pseudo-Source Level, Telia Research AB, Technical Report 13/0363-04/FCPA 109 0001, 23. May 1997.

[4] D-ETR SMG-50402 v0.9.3: 4/1997, Annex 2,