Statistical analysis of RTT variability in GPRS and UMTS networks

Jorma Kilpi* and Pasi Lassila**

*VTT Information Technology, P.O.Box 12022,
FIN 02044 VTT, Finland.

**Helsinki University of Technology, P.O. Box 3000,
FIN 02015 HUT, Finland.

Email: Jorma.Kilpi@vtt.fi, Pasi.Lassila@hut.fi

November 18, 2005

Abstract

Abstract: We study the data from a passive traffic measurement representing a 30 hour IP traffic trace from a Finnish operator's GPRS/UMTS network. Of specific interest is the variability of the Round Trip Times (RTTs) of the TCP flows in the network. The RTTs are analysed at both micro- and macroscopic level. The microscopic level involves detailed analysis of individual flows and we are able to infer various issues affecting the RTT process of a particular flow, such as changes in the rate of the radio channel and the impact of simultaneous TCP flows at the mobile device. Also, Lomb periodograms are used to detect periodic behavior in the applications. In the macroscopic analysis we search for dominating RTT values first by Energy Function Plots and then by exploring spikes of empirical PDFs of aggrgate RTTs. Comparison of empirical CDFs show significant shift in quantiles after considerable traffic increase which is found to be due to bandwidth sharing at individual mobile devices. Finally, we investigate further the impact of bandwidth sharing at the mobile device and show how the increased amount of simultaneous TCP connections seriously affects the RTTs observed by a given flow, both in GPRS and in UMTS.

1 Introduction

Traffic measurements are an important means when assessing the performance of real networks. Using measurements one obtains direct evidence of the performance and, by examining the measured data more closely, it is also possible to detect bottlenecks and/or anomalous behavior in the networks. Technology development is presently especially rapid in the field of wireless data networks. Hence, measurements in such networks, for example GPRS networks or newly deployed UMTS networks, are critical and are needed to understand the issues affecting the performance of TCP/IP traffic in such networks.

A measurement from a real operational network is always difficult, both technically and legally. Technically, one major methodological issue is whether the measurement process/method is active or passive. An active measurement process allows one to carefully design the measurement such that the phenomenon of interest can be examined very accurately. However, one major drawback of the approach is that it is intrusive in the sense that the extra traffic caused by the measurement may impact the results. Thus, there is always a fundamental question, whether a tailored active measurement is actually representative of the real network traffic. Passive measurements, on the other hand, refer to a measurement where one simply measures (captures) the real network traffic without injecting any extra traffic, and the method is hence by nature non-intrusive. Arranging such measurements is usually more difficult, requiring sophisticated measurement equipment and negotiations with network operators. The benefits are that there is no doubt about the representativeness of the data and one can try to find out whether some phenomenon exists at a frequency that is statistically relevant.

In this paper we study the data from a passive non-intrusive traffic capture measurement representing a 30 hour IP traffic trace measured from one GGSN element of a GPRS/UMTS network in a major Finnish network operator's production network. TCP is the most often used transport protocol in IP networks and the objective of the measurement is to analyze certain properties of TCP flows where one end point of the flow is in the GPRS/UMTS network and the other in the public Internet. Due to the structure of the GPRS/UMTS network, measuring at the GGSN node gives the possibility to observe all TCP flows which traverse the considered GGSN node in the network. Of specific interest in the data is the variability of the Round Trip Times (RTTs) observed by the TCP flows. Large variability of the RTTs may cause problems for example to TCP's retransmission mechanism in the form of spurious timeouts (see, e.g. [10], [8], [7], [18], [12] and [16]). As opposed to the active measurement approach used in [10] and [8], we have a passive measurement of the actual network traffic (comments on other passive measurement studies are provided Section I.A).

The objective of this study is to analyse thoroughly the statistical properties of the RTT process as observed by the TCP flows. To this end, we perform a detailed analysis of the RTTs of selected individual flows (microscopic analysis) and the RTTs of the aggregate traffic (macroscopic analysis). Furthermore, as the measurement data contains TCP flows both from GPRS users and UMTS users, we are able to compare the properties of the RTT process for both technologies.

Briefly, our measurement analysis methodology and our main observations are the following. First the quality of the time stamps on the captured IP packets is validated and we demonstrate that the measurement equipment or the setup itself has not impaired the data. The summary data is obtained by using a program called `tstat` [1], which gives information about TCP connections/flows. We use the summary data to get information about the raw data and, with the help of summary data, select TCP flows that seem to be interesting for our microscopic-level analysis. The idea is to select long flows, so that it is possible to detect qualitative changes in the RTT process over a relatively long period of time. In the microscopic analysis, we are able to address issues such as changes in the sending rate of a TCP flow and the impact of simultaneous TCP connections. Furthermore, by using the Lomb periodogram (see Section I.A), one can identify if transmission patterns behind the RTT process are periodical. In the macroscopic analysis we search for dominating RTT values first by Energy Function Plots, based on the Haar wavelet, and then by exploring spikes of empirical PDFs of aggregate RTTs. Comparison of empirical CDFs show significant shift in quantiles after considerable traffic increase which is found to be due to bandwidth sharing at individual mobile devices. Additionally, we show how the change of tariff has a clear impact on the way that the users behave - during the flat rate period users seem to open more simultaneous TCP connections than during the day time indicating increased web activity during flat rate charging. Finally, the importance of the amount of simultaneous TCP connections a mobile has, both in our micro- and macroscopic analysis, on the maximum RTTs a flow observes, lead us to believe that it is actually the presence of competition for bandwidth at the mobile device that is responsible for the very high values of RTTs that some flows observe. This issue is investigated in detail and, indeed, we show that increased degree of bandwidth sharing results in larger (even excessive) maximum RTTs.

The paper is organized as follows. In Section II, the measurement setup and the equipment is discussed, and the quality of the time stamps is verified. The microscopic analysis of certain flows, both for GPRS and UMTS, are presented in Section III. The macroscopic analysis of the RTTs is in Section IV, while Section V analyses the impact of bandwidth sharing at the mobile device on the RTTs. Conclusions are provided in Section VI.

1.1 Related Work

Many measurements of RTT have been recently made both in mobile/wireless and wired access networks. For example, a similar GPRS measurement made in Austria is reported in [17] and a study of spurious timeout events in TCP/GPRS is reported in [18]. In addition to this, a similar GPRS measurement made from seven different European and Asian countries is reported in [4]. Our study complements these studies by representing a GPRS traffic measurement made in Finland, and its statistical analysis.

Our emphasis has also been on investigating existing statistical tools that could be used in analysing RTTs. The paper [3] reports a similar, but non-mobile, study of RTT variation based on a very large data set collected at the ISP link of the University of North Caroline at Chapel Hill. Their data contained also some, but unknown, proportion of wireless access connections. They defined a connection experiencing 'high variability' if *the interquartile range* (IQR), *i.e.* the difference between $75th$ and $25th$ percentiles of an RTT sample of the connection, was strictly larger than the difference between the median ($50th$ percentile) and the minimum,

$$IQR > \text{ Median RTT} - \text{ Minimum RTT}. \tag{1}$$

Minimum RTT is used as the estimate of the non-random component of RTT. In the data they had the condition (1) accounted for over 50% of the connections. Use of percentiles makes the statistic

$$IQR/(\text{ Median RTT} - \text{ Minimum RTT}), \tag{2}$$

briefly IQR/(Md-Min), very robust but requires the number of the RTT sample values per flow to be large enough. In [3] the authors required at least 10 valid RTT samples per flow.

The paper [9] used *wavelets*, more precisely *energy function plots* (EFPs), applied to the 'signal' of aggregate traffic, both measured and simulated, in order to detect network performance problems indicated by significant changes in the dominating RTT values.

In [15] a statistical tool called *the Lomb periodogram* (LP) [11] was used in a very interesting experimental set-up where it was able to find timing properties of a data flow from the trace that contained only traffic of another data flow. This latter flow had only shared some resources with the former flow.

We have also tested and used (2), LPs and EFPs as a methodological starting points in our analysis.

2 Measurement environment and equipment

2.1 The monitoring port of a GGSN

The measurement point was the monitoring port of the Gi interface of one of the GGSN elements of the operator's GPRS backbone, see Figure 1. Both the switch, where the monitoring port was, and it's operating system were standard products of a well-known vendor. Connected to this monitoring interface there was a 100 Mb/s Ethernet which enabled the use of standard packet capturing methods, such as `tcpdump`. Between the measurement point and the Internet access there is a small operator's service network, which contains the Network/Private Address Translation (NAT) box and the firewall router.

2.2 A rough preview of the traffic

Currently, mobile data in this network consists mostly of traffic from plain GSM/GPRS users and, in suburban areas, also from EDGE and UMTS users. The operator had launched

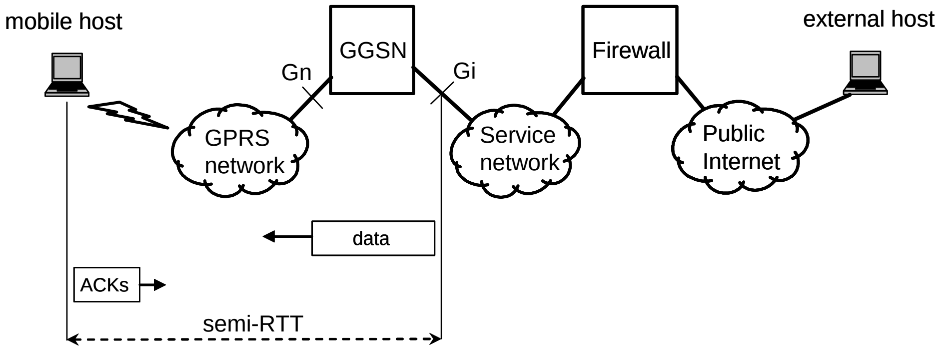

Figure 1: The measurement scenario.

commercial UMTS services in major cities in November, 2004. Some of the subscribers of the operator have only volume based charging but some portion of the subscribers had also flat rate between 18.00-06.00 during working days and on weekends. The effect of this *flat rate population* is clearly significant, see Figure 2. Figure 2 has been obtained from the SNMP data of several GGSN elements and depicts five minute averages of bits per second for a 32 hour period from Wednesday, November 10th, 2004, at 9:30 to Thursday, November 11th, 2004, at 16:30.

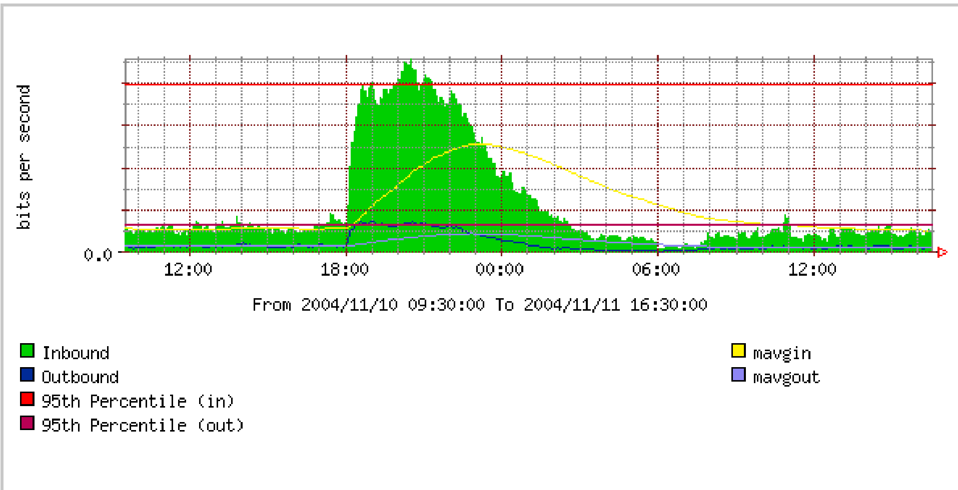

Figure 2: The SNMP data from MIBs shows that overall daily traffic profile is significantly affected by the subgroup of users who pay flat rate between 18:00-06:00.

Our measured data trace was obtained from one GGSN, hence it does not contain all of the simultaneous GPRS traffic of the whole access network. However, the daily traffic profile at any of the GGSNs of the operator is similar to one in Figure 2. This is because new users are not connected to a certain GGSN by regional basis but according to the time instance of the PDP context activation. There is load balancing between all GGSN elements in the sense that the fraction of users connected to one GGSN element is, in the long run, relative to the total number of GGSNs.

2.3 Technical requirements

Capturing of TCP traffic at the above described measurement point for a long enough time, say 24 hours, should give a representative sample of TCP connections. The remaining technical requirements are the quality of time stamps, as all RTT statistics are based on them, and capturing of every single packet.

Minimum values of IP packet transfer delay is hundreds of milliseconds for GPRS/GSM and tens of milliseconds for UMTS. Thus, millisecond accuracy would be sufficient. However, to guarantee any accuracy is not easy to achieve using Linux, `tcpdump` and an ordinary off-the-shelf PC.

The time stamps are called *precise* if their random variability is small. The time stamps are called *accurate* if both their variability *and* bias are small. The time stamp that the `tcpdump` program gives is associated to the end of the corresponding packet, see [14]. The bias is thus considered as relative to the 'time instance of the last bit of the Ethernet frame that carried the corresponding packet'.

Since we needed to capture and save all the traffic in a disc, while the quality of time stamps must simultaneously be kept high, some optimisation of the measurement equipment, a 2.4MHz computer with Linux operating system and a 160 GB IDE hard disk, had to be made. The computer has the 10/100 Ethernet card integrated in the mother board. We used Debian Linux kernel 2.6.8 compiled with *the preemptible kernel* option. The hard disk was optimised using options that enabled *Ultra Direct Memory Access* (UDMA) mode, 32 bit words and simultaneous writing onto 16 sectors with one CPU command. The file system, `ext3`, was optimised using *synchronous* writing and minimising journaling overhead. Following the advice from [5] the measurement was performed in the single-user mode with NTP[1] disabled.

Due to SNMP information, like in Figure 2, and a preliminary test measurement we had quite precise knowledge about the traffic load at the measurement point. The packet capturing and time stamping features of the optimised Linux computer, described above, were tested beforehand using Ethernet 10/100BASE-T and GbE interfaces of the Adtech AX/4000 traffic generator of Spirent Communications and with a load that was two and three times more than the observed maximum load at the actual measurement interface was. Even with a threefold load the packet capturing, time stamping and saving still worked perfectly. In fact, the maximum bias with the traffic generator test with threefold load was less than our target accuracy $\pm 50\mu s$. This is about the best that can be achieved using ordinary NICs, since they do not generate CPU interrupts more often.

The data trace we use in this paper was measured from Wednesday, November 10th, 2004,

[1] NTP was implemented only to see the drift of the computer's clock, -1.21 ± 0.5p.p.m.

at 10:30 to Thursday, November 11th, 2004, at 16:30. The `tcpdump` program reported 0 dropped packets, and the Ethernet card's driver did not report any dropped or anomalous Ethernet frames.

One way to check the accuracy of time stamps is to plot the packet inter-arrival time $t_{i+1} - t_i$ against the packet's size B_{i+1}, remembering that the time stamp of `tcpdump` must be associated at the end of the packet's Ethernet frame. This is shown in Figure 3 below: with accurate time stamps we should be able to see the 100 Mb/s Ethernet's transfer rate which, knowing the framing structure of the Ethernet, can easily be calculated theoretically assuming packets arrive back-to-back. The calculated dashed line is also shown in Figure 3. The packets' true spacings cannot be too close to each other. If they are too close to each other, it is a sign of buffering at the measurement equipment before the time stamp is attached to packets. Figure 3 represents a piece from the busy hour period and points below the dashed line show that some buffering before time stamping had occured. The bias for large packets is the vertical distance between a point and the dashed line. As can be

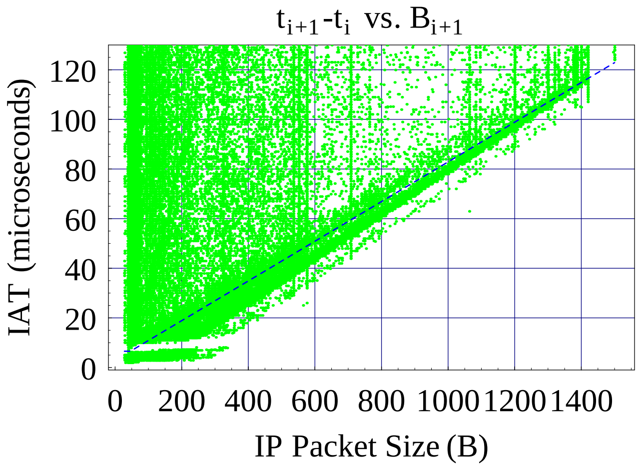

Figure 3: Packet inter-arrival time against the packet size.

seen, the granularity of time stamps is not sufficient to distinguish correctly small packets that arrive back-to-back. Their inter-arrival time may be too small but is within the $\pm 50\mu s$ bound. This is not a problem in our study.

Another way to view the time stamps t_i is to make plots of $(t_i, t_i \mod h)$, with a suitably chosen h, for example, $h = 1ms$ or $h = 1s$. These type of plots show if there are significant gaps in the time stamps. Such gaps do not show up in plots like 3. In summary, we are confident that the accuracy of time stamps is very good with bias less than the target accuracy $\pm 50\mu s$.

2.4 Re-building of TCP connections

Before anonymization the packet level data was given as input to the `tstat`-program, [1]. We used a test version 1.0beta15 of `tstat`. When processing the packet data, `tstat` maintains a list of all TCP connections and, when analysing the next TCP/IP packet, rebuilds the corresponding TCP connection status. If all the phases of a TCP connection are properly observed, the connection statistics are calculated and written into a file as one record which contains 92 different attributes. Otherwise a record is written into another file. The record in the file of incompletely observed connections still contains valid information about IP-addresses, TCP ports and the number of SYN packets. It was used to detect anomalies, like port scanning, in the data. However, all our analysis in this paper is based only on those connections that `tstat` recorded as *completely observed* in the sense that opening and closing were correctly observed, as described above. Since the packet loss in the measurement was zero, these completely observed connections are complete also in the sense that every single packet of them is also observed.

2.5 Semi-RTT

Semi-RTT refers to the difference of the time stamps of a (downstream) TCP/IP packet carrying some data payload and of the corresponding ACK packet, see again Figure 1. Since these time stamps are measured at the Gi interface the semi-RTT is not the same as an end-to-end RTT but, like in [17] we call it simply as RTT.

For each completely observed TCP connection `tstat` provides some (semi-) RTT information automatically: minimum, maximum, mean, standard deviation and the number of times a data segment and the corresponding ACK has been observed. This latter value, denoted here by *RTT count*, is directly relative to the size of the flow. Moreover, to avoid that out-of-sequence and duplicate segments or ACKs provide unreliable measurements that affect the RTT measurement itself, only samples referring to in-sequence segments are considered. This RTT information is given in both down- and upstream directions.

However, since the built-in RTT statistics of `tstat` were not enough for this study the code of the `tstat`-program was modified in a way that we got out every valid RTT value of every completely observed connection and enough information to associate each individual RTT value with the connection that it belongs to.

3 Classification methodology

The *arrival time* of a completely observed TCP connection is determined by the time stamp of the handshaking SYN packet sent by the client. Downloadings are first classified according to TCP connection arrival times into 5 distinct groups as explained in Table 1 below. Recall that the flat rate population is a subset of subscribers of the operator.

Sample Number	TCP Connection Arrival Time	Flat Rate Population Present?
1	10:30-18:00	No
2	18:00-21:00	Yes
3	21:00-24:00	Yes
4	00:00-06:00	Yes
5	06:00-16:30	No

Table 1: Division into subsamples according to TCP connection arrival time.

Samples 1 and 5 were statistically similar to each other, also samples 2 and 3 were similar to each other. Thus, to analyse the impact of tariff change, we compare only sample 1 versus 2. Analysis of the sample 4 is not contained in this paper.

3.1 RTT Count as a measure of a size of the downstream flow of a TCP connection

Further division of data is based on the concept of the (downstream) RTT Count which is the number of valid RTT samples per downloading part of a TCP connection. Since the effect of the packet size is significant, the RTT value obtained from the three-way handshake is not included in our analysis.

RTT Count is a measure of the size of the downstream flow of a TCP connection. It is relative both to the duration of a downloading and to the amount of (unique) bytes that are downloaded, see Figures 4 and 5. The same concept was essentially used in [17] as a measure of size.

The concept of *Unique Bytes* is defined as the amount of bytes correctly acknowledged, not including retransmitted, duplicated or in some other sense anomalious bytes. This is also one of the statistics that `tstat` provides.

Figure 4 shows RTT Count compared against downloaded Unique Bytes (MB). The lines show how the size in bytes would develope if each increment in RTT Count would be due to acknowledging only a single segment of size 536 or 1460 bytes, respectively.

For a downloading flow the RTT count increases approximately linearly with Unique Bytes. If the corresponding TCP connection downloads data in small segments the amount of downloaded Unique Bytes can be small with RTT count large. If, for example, a connection generates other TCP connections for data transfer, its duration can be very long even if the downstream RTT count is rather small as shown in Figure 5.

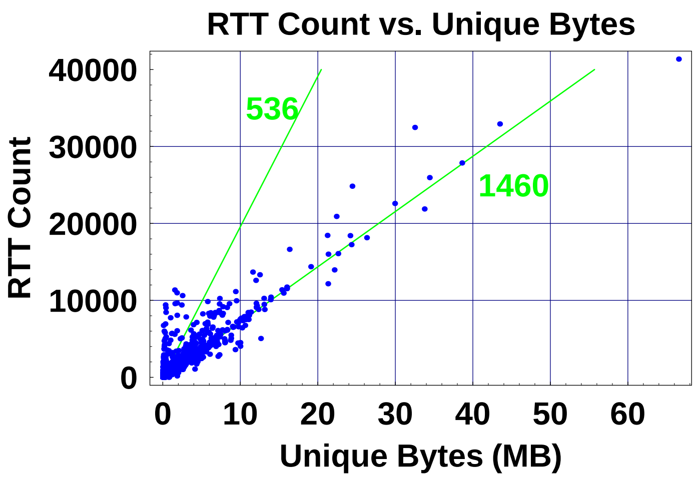

Figure 4: RTT Count against downloaded Unique Bytes (MB).

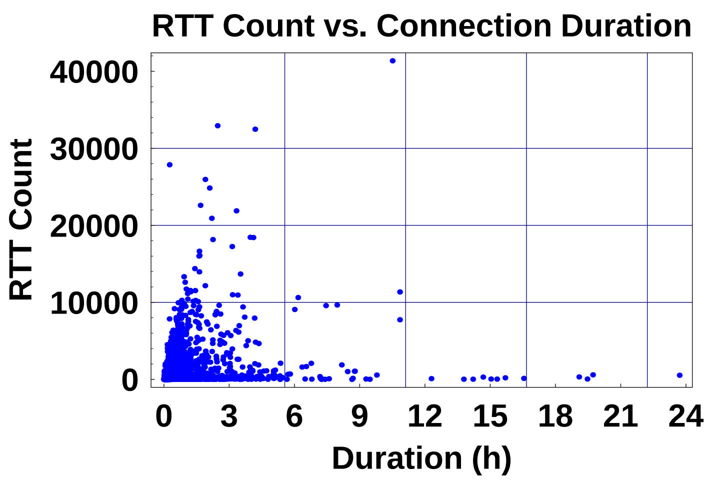

Figure 5: RTT Count against TCP connection duration.

3.2 Division of downstream flows according to RTT count

We are not very interested in the duration of a flow. The word 'long' is used here to mean long in the sense that the RTT Count is large. More detailed statistical analysis is made only for those flows that were longer than at least 6 RTT counts, with RTT from the handshaking exluded. Figure 6 shows that using the value 6 as a discriminating value divides each sample

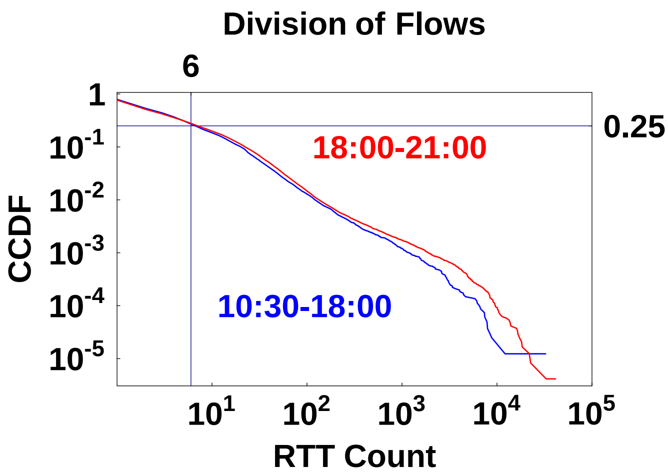

Figure 6: The condition RTT Count $\geq$ 6 includes $\approx$ 25% of the downstream flows (and excludes $\approx$ 75%).

into approximately 25% of long flows and 75% of short flows. The tariff change does not have any effect on that. Flows shorter than 6 RTT counts were analysed only by studying the distribution of the range of RTT. This is presented in section 5.1.

We also studied the number of applications with relatively long flows in terms of RTT Count. Table 2 shows, according to the server port number, the 12 most frequent applications which had RTT Count $> 1\,000$. In total, there were 145 different applications that had RTT Count $> 1\,000$.

4 Microscopic analysis of long TCP flows

A rather complete analysis of a number of individual long flows were primarily made in order to understand how the radio interface affects the RTTs. There were also other questions, such as does the considerable increase of traffic due to the tariff change show up somehow in RTTs? The question that concerns the operator is that does the flat rate population disturb ordinary subscribers who pay per volume? (The flat rate population is not directly related to the operator but through a household appliance selling company.) In general, the idea is to find out what can be deduced from changes in the RTT during a flow.

Some specifically chosen examples are presented here first in order to give a reader a flavour

Port	Application	Explanation	Frequency
119	nntp		9
1863	msnp		9
443	HTTPS	Web	11
4662	eDonkey	P2P	11
1214	kazaA	P2P	13
411	rmt		13
43594	runescape	Game	22
8080	HTTP-alt	Web	24
10000	ndmp		25
110	pop3	Mail	30
6699	napster?	P2P	46
80	HTTP	Web	439

Table 2: The 12 most frequent applications that had RTT Count > 1 000.

of what is behind a macroscopic level analysis that will be presented later. Example flows were chosen because the time series

$$\{(t_i, RTT(t_i)) \mid i = 1, \ldots, \text{RTT Count}\} \tag{3}$$

had some distinctive features. In (3) the t_i is the time stamp of an ACK packet and $RTT(t_i)$ is the RTT value calculated from the ACK packet.

4.1 Individual TCP flows in GSM/GPRS

Some key properties of the three individual TCP/GPRS (and two TCP/UMTS) downstream flow examples are collected to Table 3 below. They all used the SACK option. Unique Bytes is given in MB.

Property	**Flow**				
	1	**2**	**3**	**4**	**5**
SACKs	2795	22	797	165	74
RTT Count	32 479	7 427	16 078	6351	13961
Unique Bytes	32.5	9.9	22.6	5.6	22.1
IQR/(Md-Min)	0.79	0.11	0.29	0.84	0.36

Table 3: Three TCP/GPRS and two TCP/UMTS downstream flows.

4.1.1 Flow 1

The first flow was chosen since we wanted to understand what caused the improvement in RTT values after about half an hour as shown in Figure 7. Another point to notice with

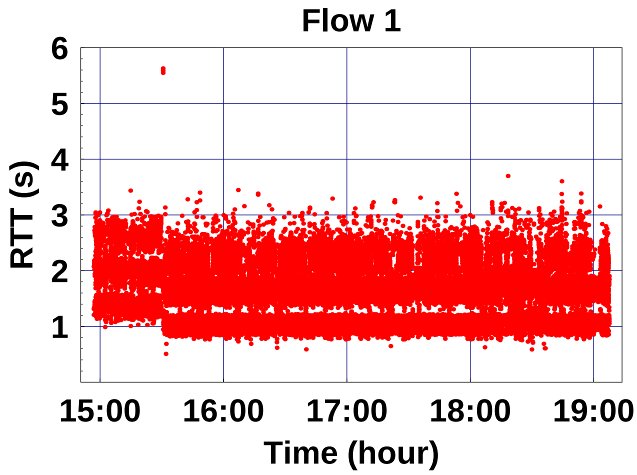

Figure 7: Time series (3) of Flow 1

Flow 1 is that the rise in traffic due to tariff change at 18:00 might have some effect for this particular flow.

Improvement in RTTs after about half an hour was easily seen to be due to change from large to a smaller segment size, from 1380 to 536 bytes. What caused this change? Flow 1 is most probably an example of a streaming application, audio, even though it was behind TCP port 80. This streaming application probably had a 3 second buffer. Indeed, Figure 8 shows a very reqular pattern not typical to TCP.

Analysis of the receiving rate of packets revealed that the source is receiving data at a constant rate of 18.2 kbit/s. Careful inspection of Figure 8 shows that data burst of fixed size are sent every 3:rd second on the average. It also takes about 3 seconds before every segment of one burst have been acknowledged by the mobile host. However, after the change in the segment size, the data burst could be sent slightly better within the 3 second intervals in the sense that the sender received ACKs slightly earlier. Thus, probably the streaming application forced the change in the segment size. Finally, another important thing to notice was that there were essentially no other simultaneous TCP connections from the same mobile.

To further analyse the flow, the LP has been computed corresponding to the beginning and end of the flow (see Figure 9). From the figure it is clear the in the beginning the burst period is slightly above 3 seconds. After the change in the segment size, the burst period is somewhat below 3 seconds. Looking closely at Figure 7 one can observe that after the tariff change at 18:00 the level of the RTTs rise again (probably due to increased traffic in the network). Computing the LP from the end shows that the burst period increases to slightly

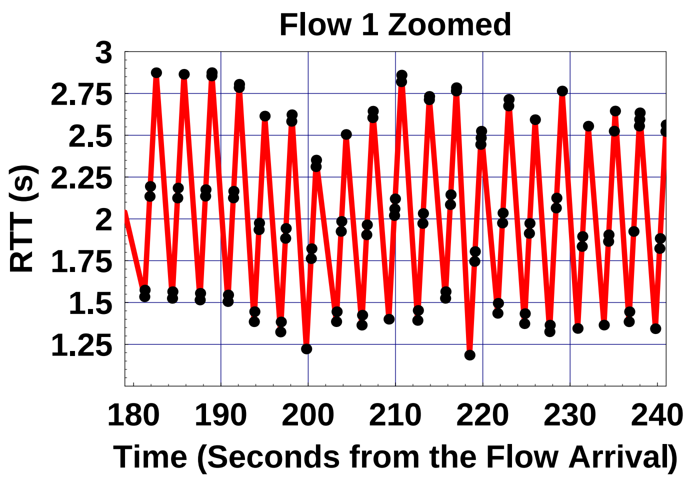

Figure 8: Time series of Flow 1 zoomed.

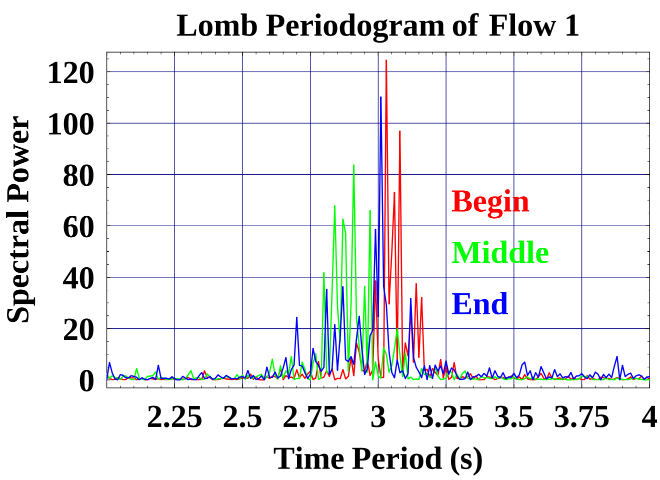

Figure 9: The Lomb Periodogram calculated for Flow 1.

above 3 seconds again.

4.1.2 Flow 2

The Figure 10 shows the remarkable property of flow 2, namely that RTTs at level 5.0-7.5 seconds seems to be normal for this connection. Also, the minimum is less than 1 second,

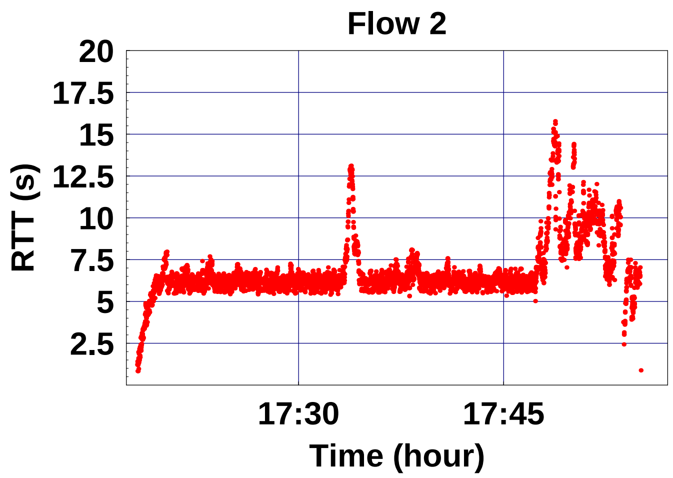

Figure 10: Time series of Flow 2

hence a better level could be possible.

Unlike for Flow 1, within the limits 5.0-7.5s use of LP showed that there was no regular periodic structure for Flow 2. Moreover, as Figure 11 indicates, there was a change in the downlink capacity from 40.2 kb/s to 26.8 kb/s, which corresponds to a loss of one PDCHs in the downlink when CS-2 is the channel coding scheme. Finally, as Figure 12 shows, the change in the link capacity for Flow 2 was due to self-congestion caused by other simultaneous TCP connections from the same mobile host. Large RTT values were probably due to a low terminal capacity.

4.1.3 Flow 3

Again, use of LP showed no regular periodicities. However, Flow 3 had some peculiar reqular intervals of bad RTTs. In this case the self-congestion is also an explanation since the user was simultaneously running an application (MSNP) which caused, each time with a new flow in port 80, regularly within approximately 8 minute intervals a downloading of exactly 265.3 kB file. These downloadings occur exactly at the bad intervals visible in Figure 13.

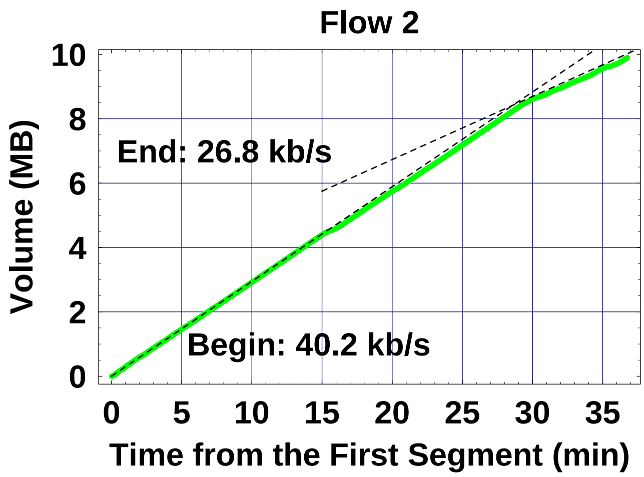

Figure 11: Change in the downlink capasity for Flow 2.

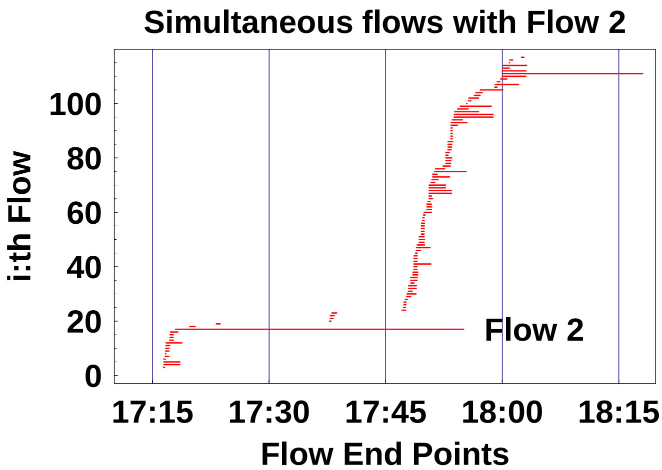

Figure 12: Change in link capasity for Flow 2 is due to self congestion.

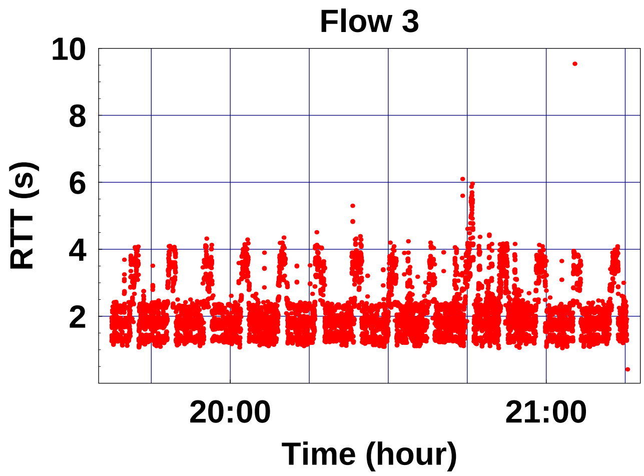

Figure 13: Flow 3: What caused so reqular intervals of large RTTs?

4.2 Individual TCP flows in UMTS

During the measurement there were only piloting UMTS usage in the network. The UMTS services were commercially launched after the measurement was done. However, the absolute number of observed TCP connections that could be associated to UMTS was sufficient to make comparative analysis against GPRS also from our data set. At best, RTTs in UMTS are about one magnitude smaller than with GPRS, see also [17].

Flows 4 and 5 were originated from the same mobile. They are both shown in Figure 14. The level of RTTs of Flow 4 increase when Flow 5 starts. There were also several other

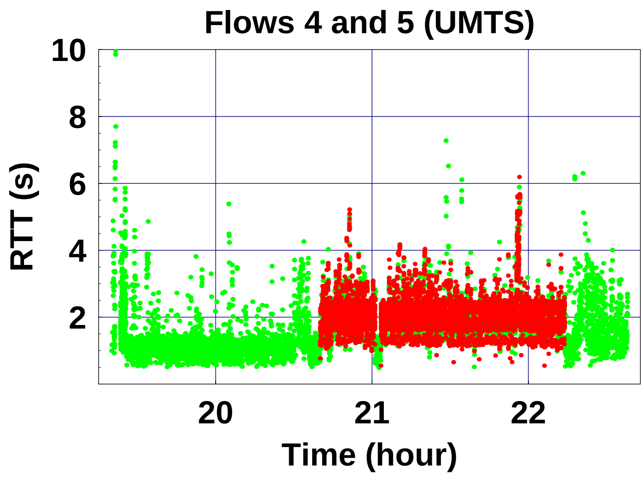

Figure 14: Time series of Flows 4 and 5.

long flows simultaneously from the same mobile. Almost all RTT variations are explained by these other flows.

5 Properties of RTT at the macroscopic level

5.1 Short downstream flows

When the RTT Count is less than 6, with the handshaking RTT not included, it is not easy to make any kind of analysis about the RTT process (3). As already mentioned these short flows consist of about 75% of TCP downstream flows, but only a fraction of all downstream Unique Bytes. For the sake of completeness we also present some analysis of these short flows.

The RTT range, difference of the maximum and the minimum RTT values per flow, seems to be the best general statistic for short flows. We know that these extremal values are highly non-robust but we trust the accuracy of our time stamps. Figure 15 shows the CDFs of RTT ranges of downstream flows of sample 1, when classified according to RTT Count with values 3,4,5 and 6. It contains also the case RTT Count = 6 in order to make it overlap with the analysis of long flows.

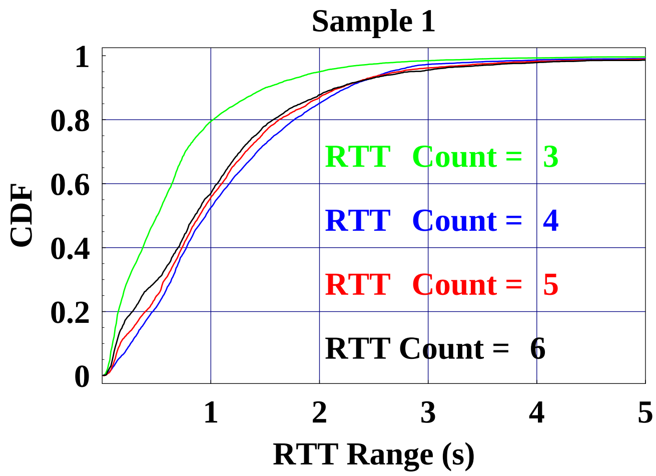

Figure 15: The range of RTTs of short flows.

Figure 16 compares short flows of samples 1 and 2. For clarity, only RTT Counts 3 and 5 are shown, others are similar. As can be seen, the ranges of the RTTs clearly increase.

5.2 The dominating RTT values

A superposition of many TCP connections can be assumed to produce local periodicities in *the packet level aggregate traffic* since the dynamics of an individual TCP connection

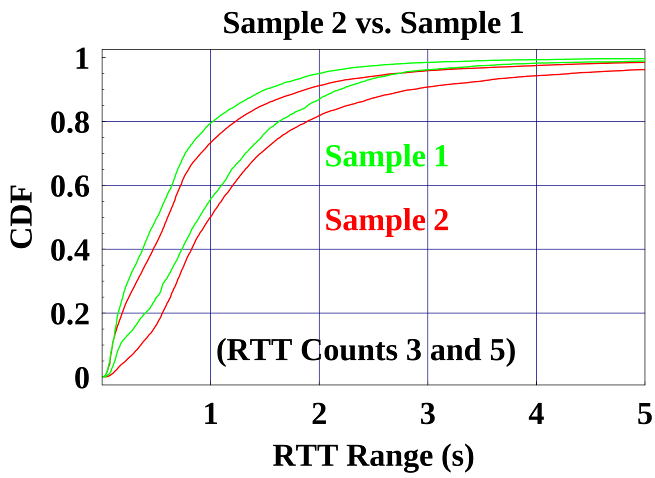

Figure 16: Comparison ranges of RTTs of short flows.

depend on RTT. If there are dominant values of RTT, they should then be 'hidden' also in the aggregate TCP traffic rate.

Figure 17 shows how the EFP, using *the Haar wavelets*, see [9, 6, 13, 2], looks for our data. It contains energy values of different scales of two packet level aggregate traffic samples where the flat rate population is present and one sample, represented by the lowest curve, from the time period where the flat rate population is not present .

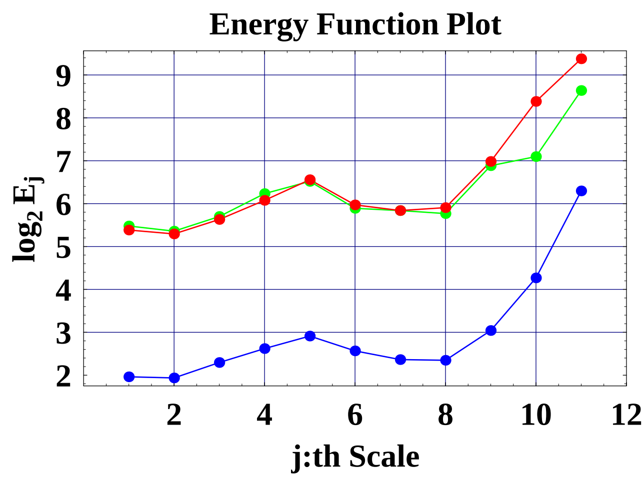

Figure 17: Energy function plot.

Significant periodicities at the corresponding scale should show up in EFPs as 'dips' downwards. The 0th scale in Figure 17 contains packets per $10ms$ bins, the dip occurs in all cases in scales 6,7, 8 and 9 which corresponds to $0.64s$, $1.28s$, $2.56s$ and $5.12s$ bins. The dominant RTT values, if they exist, can be expected to be found between half second and five seconds.

In order to see more closely the values where the dips downwards starts and where the curves start to rise up again we simply varied the initial bin sizes. We made this for samples where the flat rate population is present and for samples where the flat rate population was not present. The differencies were found small.

We now turn back from the packet level data to the RTT data. From now on the word ’aggregate‘ in this section means the set of all RTT samples without making any distinction to what flows they belong. The long flows, with RTT Count at least 6, are not separated from short flows either.

In order to compare dominating RTT values we calculated estimates of the PDFs simply as normalised histograms. Normalised because we wanted to compare the positions of spikes in the histograms. The heights of the spikes are then only relative to to sample sizes, which were different. Sample 2 was about four times larger than Sample 1.

In GPRS, the Abis interface causes a granularity of 20 ms in the upstream traffic. Smaller variations are basically not due to radio interface. We wanted to capture the variations due to radio interface but we were not so interested in the variations caused by the network. Hence we decided to use a bin size of 10 ms.

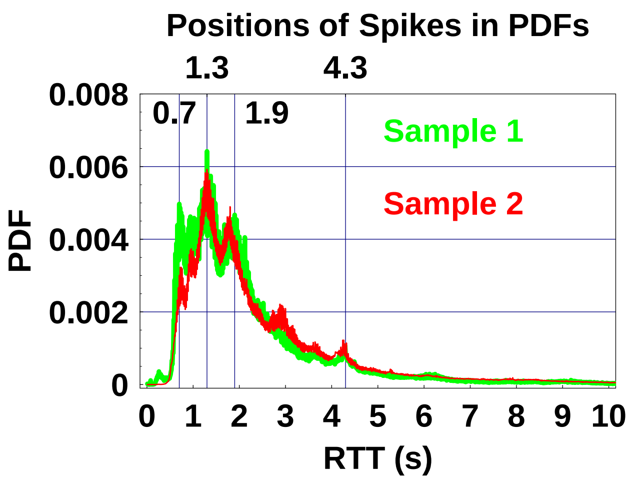

Figure 18: The positions of spikes do not differ much.

The long flows (25% of all) dominate the histograms: If the short flows (75% of all) were left out, the histograms would still have the same shapes and spikes at the same positions. On the other hand, leaving out the longest flows of each sample does not change the histograms either. The Sample 1 had spikes at positions 0.7, 1.3, 1.9 and 4.3 seconds, they are marked in Figure 18. The Sample 2 had also spikes at the same positions. Note that the positions of the spikes correspond to the typical transmission delays of a packet and its acknowledgement when the packet is sent directly without any buffering at SGSN where the buffer is per mobile, after a queuing delay of one data packet, etc. This also implies, that the long flows are not much affected by other traffic in the network. They are only constrained by their own limited radio access link.

The prior information about lower tariff population, that some of them use mobile as an alternative to dial-up connections in the regions where no broadband is available, suggests that typically they are far from base stations also, hence they may also often have heavier FEC property in their channel codings than the normal tariff population. Indeed, sample 2 has clear spikes also in about 1 second and slightly less than 3 seconds in places where the sample 1 does not have so clear spikes.

The positions of spikes do not practically differ with different samples so histograms of samples 3 and 5 need not be shown here. Especially the traffic increase due to tariff change does not seem to have significant effect on positions of spikes. Only for the relative heights of the spikes. This suggests that, in case of long flows, perhaps the lower tariff population does not disturb normal tariff population significantly.

The Flow 2 in Figure 10 shows that minor spikes in the histogram of sample 1 near 6 seconds are still in the range of our interest.

5.3 Changes in quantiles

From the histogram based estimates of the PDFs we calculated also CDFs. Long flows dominate also them. Figure 19 compares samples 1 and 2. Shift in the 0.25 and 0.5

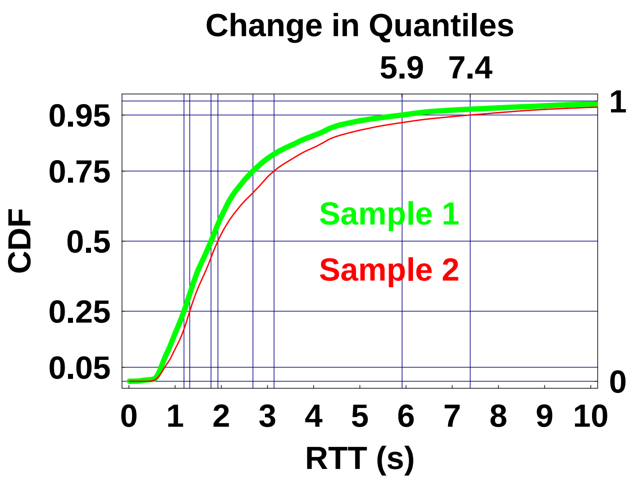

Figure 19: Comparison of CCDFs.

quantiles is likely due to increased amount of long connections that are not close to base stations and thus use heavier channel coding scheme.

Significant increase of the 0.95 quantile, from 5.9s to 7.4s, is due to the increased amount of bandwidth sharing at individual mobile device due to simultaneous TCP connections from it. Indeed, in Figure 20 we have calculated the maximum number of simultaneous TCP connections observed from the same mobile. Before 18:00 50% of mobiles had at most

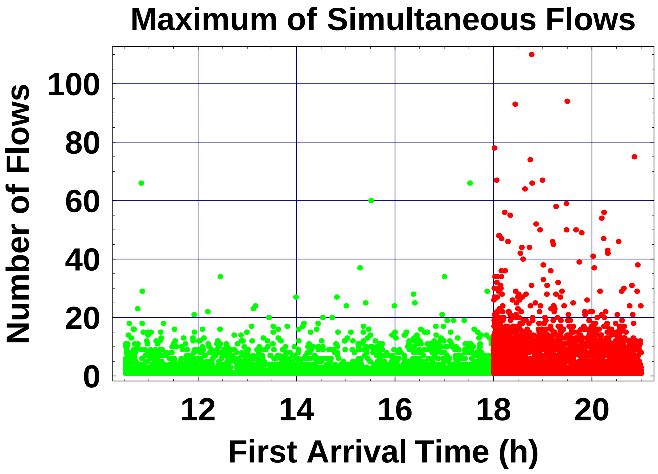

Figure 20: Maximum number of simultaneous TCP connections from the same mobile.

one TCP connection at time and 50% had at least two simultaneous TCP connections, but during 18:00-21:00 50% of mobiles had more than 5 simultaneous TCP connections active at least once in their whole Internet session.

5.4 Self-congestion

In order to further study how simultaneous TCP connections from the same mobile actually affect the RTTs, we divided such GPRS sessions, that had at least two simultaneous TCP connections, into two groups. The Group 0 did not send much data (Unique Bytes) into upstream direction whereas Group 1 contained those that must have had at least one non-trivial data transfer in the upstream direction. More precisely, Group 1 was defined as the set of those mobiles for which the total amount of upstream Unique Bytes was larger than 1 kB times the total number of TCP connections from that mobile. Figures 21 and 22 show the maximum RTT over all of the TCP connections of a mobile against the maximum number of simultaneous connections observed from the same mobile.

The scales of axes are chosen to be the same for both Figures 21 and 22 in order to show that extremely large RTT values occur almost solely for group 1, whereas very large number of simultaneous TCP connections occur for group 0.

It can also be expected that observed maximum RTTs increases as the amount of simultaneous TCP connections increases. This is verified in Figure 23, which shows a robust estimate of expectation of maximum RTT over all simultaneous flows, conditioned over the maximum number of simultaneous connections.

Our UMTS sample is not large enough to make the same division as with GPRS, but Figure

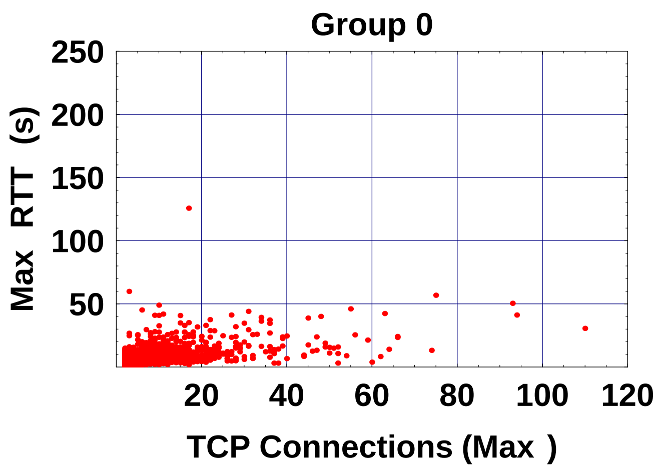

Figure 21: No significant upstream traffic.

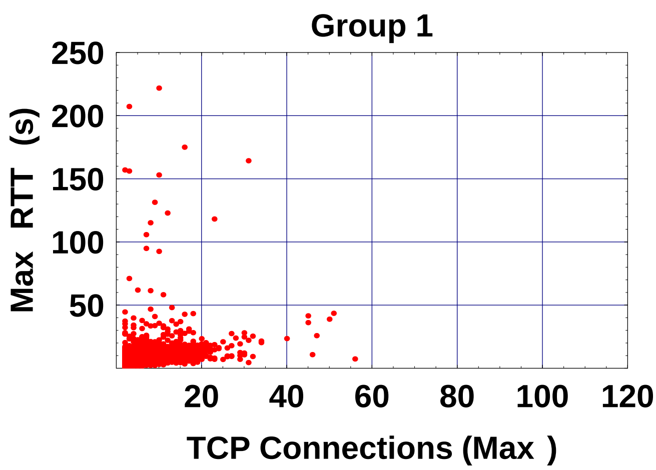

Figure 22: Significant upstream traffic.

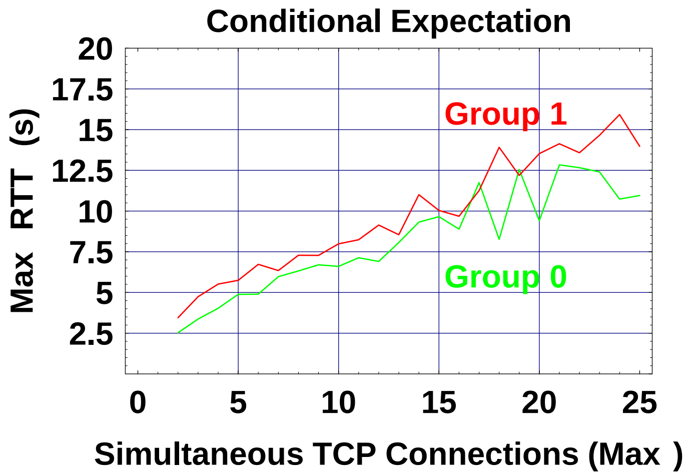

Figure 23: Robust estimates of expectations.

24 shows a picture similar as 21 and 22, but now for all UMTS mobiles. Note that the vertical axes of Figure 24 have a different scale than in Figures 21 and 22. Self-congestion is also a problem for UMTS mobiles, although slightly less serious. Figure 24 can be slightly biased

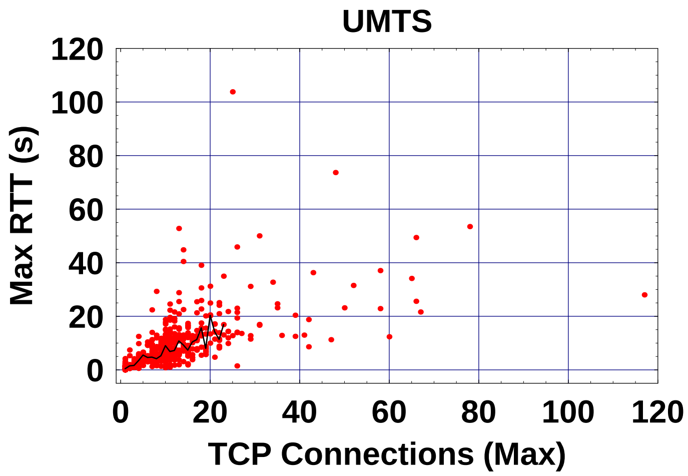

Figure 24: UMTS mobiles.

due to piloting (testing) UMTS usage.

6 Conclusions

We studied the data from a passive traffic measurement representing a 30 hour IP traffic trace from a Finnish operator's GPRS/UMTS network. Of specific interest was the variability of the Round Trip Times (RTTs) of the TCP flows in the network. The RTTs were analysed at both micro- and macroscopic level. The microscopic level involved detailed analysis of individual flows and we are able to infer various issues affecting the RTT process of a particular flow, such as changes in the rate of the radio channel and the impact of simultaneous TCP flows at the mobile device. Also, Lomb periodograms were used to detect periodic behavior in the applications. In the macroscopic analysis we searched for dominating RTT values first by Energy Function Plots and then by exploring spikes of empirical PDFs of aggrgate RTTs. Comparison of empirical CDFs showed significant shift in quantiles after considerable traffic increase which was found to be due to bandwidth sharing at individual mobile devices. Finally, we investigated further the impact of bandwidth sharing at the mobile device and showed how the increased amount of simultaneous TCP connections seriously affects the maximum RTTs observed by a given flow, both in GPRS and in UMTS.

6.1 Discussion about statistical methods

The problem with the IQR/(Md-Min) statistic is that the large RTT values are not taken into account. If, for example, every 10th RTT is 10 seconds while others are about 1 second, a user notices it but the statistic does not. Values in Table 3 were all less than 1 since major factor of RTT variability was non-random in nature.

The Lomb periodogram, applied to the RTT process (3), was found possibly useful in detecting performance of streaming applications over TCP as the analysis of Flow 1 showed. For an ordinary, non-streaming, TCP downloading the LP typically do not indicate any regular periodicity. Hence it could be used to detect streaming applications behind port 80. For example, whenever LP detects something significantly and regularly periodic, a streaming application may be a cause. The periodicity could be found out also in other ways. The point is that LP is systematic: given (3) as input it produces a clear picture as output. Or, if some significance level is given, it could classify long flows in port 80 into possibly streaming applications or definitely not streaming applications.

The way we applied EFPs was extremely straightforward. Again the point in EFP is that making of Figure 17 and varying the initial bin size requires much less work than making of Figure 18 always requires. Figure 17 contained essentially the same information about significant values of dominating RTTs as Figure 18.

Acknowledgement

The authors would like to thank Vesa Antervo from Elisa for his efforts on making and anonymising the measured trace, Jussi Vähäpassi from Elisa for financial support and Marco Mellia for his help about `tstat`.

References

[1] *Tstat:TCP Statistics and Analysis Tool*. `http://tstat.tlc.polito.it`.

[2] P. Abry and D. Veitch. Wavelet Analysis of Long-Range-Dependent Traffic. *IEEE Transactions on Information Theory*, 44(1):2–15, January 1998.

[3] J. Aikat, J. Kaur, F. Donelson Smith, and K. Jeffay. Variability in TCP Round-trip Times. In *Proceedings of IMC'03*, pages 279–284, Miami Beach, Florida, October 27-29 2003. ACM.

[4] P. Benko, G. Malicsko, and A. Veres. A Large-scale, Passive Analysis of End-to-End TCP performance over GPRS. In *INFOCOM 2004*, 2004.

[5] J. Cleary, S. Donnelly, I. Graham, A. McGregor, and M. Pearson. Design Principles for Accurate Passive Measurement. In *Passive and Active Measurement Workshop PAM-2000*, Hamilton, New Zeeland, April 3.-4. 2000.

[6] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.

[7] A. Gurtov and R. Ludwig. Responding to spurious timeouts in TCP. In *Proceedings of INFOCOM 2003*, San Fransisco, California, USA, April 2003.

[8] A. Gurtov, M. Passoja, O. Aalto, and M. Raitola. Multilayer protocol tracing in a GPRS network. In *Proceedings of the IEEE Vehicular Technology Conference*, September 2002.

[9] P. Huang, A. Feldmann, and W. Willinger. A non-intrusive, wavelet-based approach to detecting network performance problems. In *Proceedings of ACM Internet Measurement Workshop 2001*, pages 213–227, 2001.

[10] J. Korhonen, O. Aalto, A. Gurtov, and H. Laamanen. Measured performance of GSM HSCSD and GPRS. In *Proceedings of the IEEE Conference on Communications*, June 2001.

[11] N.R. Lomb. Least-squares frequency analysis of unequally spaced data. *Astrophysics and Space*, 39:447–462, 1976.

[12] R. Ludwig and R.H. Katz. The Eifel algorithm: making TCP robust against spurious retransmissions. *ACM Computer Communications Review*, 30(1), January 2000.

[13] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1998.

[14] S. McCanne and V. Jacobson. The BSD Packet Filter: A New Architecture for User-level Packet Capture. In *1993 Winter USENIX Conference*, San Diego, CA., January 25.-29. 1993.

[15] C. Partridge, D. Cousins, A.W. Jackson, R. Krishnan, T. Saxena, and W.T. Strayer. Using Signal Processing to Analyze Wireless Data Traffic. In *ACM Workshop on Wireless Security*, Atlanta, Georgia, USA, September 29 2002. ACM.

[16] P. Sarolahti, M. Kojo, and K. Raatikainen. F-RTO: an enhanced recovery algorithm for TCP retransmission timeouts. *ACM Computer Communications Review*, 33(2), April 2003.

[17] F. Vacirca, F. Ricciato, and R. Pilz. Large-Scale RTT Measurements from an Operational UMTS/GPRS Network. In *Submitted*, 2005.

[18] F. Vacirca, T. Ziegler, and E. Hasenleithner. Large Scale Estimation of TCP Spurious Timeout Events in Operational GPRS Networks. In *COST 279*, 2005.