End-to-end measurements over GPRS-EDGE networks

Juan Andrés Negreira
Facultad de Ingeniería,
Universidad de la República
Montevideo, Uruguay

Javier Pereira
Facultad de Ingeniería,
Universidad de la República
Montevideo, Uruguay
javierp@fing.edu.uy

Santiago Pérez
Facultad de Ingeniería,
Universidad de la República
Montevideo, Uruguay

Pablo Belzarena
Facultad de Ingeniería,
Universidad de la República
Montevideo, Uruguay
belza@fing.edu.uy

ABSTRACT

In the last years, QoS (Quality of Service) parameter estimation has become a main research area in networking due to a continuous growth of the Internet. End-to-end active measurement is one of the topics that focuses on this research area. However, these measurement methodologies have focused on end-to-end measurements over the wired Internet. The development of cellular data services is shifting the resarch focus on QoS from wired to wireless networks. End-to-end measurement methodologies of cellular networks have some issues that are not considered by traditional measurement techniques. This paper analyzes these issues and suggests an end-to-end active measurement methodology that deals with these particular problems. The proposed research and measurement methodology is based on a GSM (Global System for Mobile Communications) network, particularly on data services on a GPRS-EDGE (General Packet Radio Service/Enhanced Data Rates for GSM Evolution) network. Several experiments in different situations have been done in a real cellular network. These experiments have tested the performance of the methodology in different data access conditions.

Categories and Subject Descriptors

C.4 [**Performance of Systems**]: Measurement techniques, Performance attributes.

General Terms

Network performance, Algorithms.

Keywords

Performance, QoS, GSM, GPRS-EDGE.

1. INTRODUCTION

Today cellular services include voice communications, email, sms, audio and video streaming and mobile TV. This situation has been generated by the increase in cellular access bandwidth. This type of services have strong constraints in performance and quality. To deal with this constraints a model that analyzes the end-to-end performance of a GPRS-EDGE connection is necessary. Unfortunately, the analytical models that have been developed to analyze the performance of cellular systems make strong simplifications. They can only be used to get a rough dimension of the system, but not to analyze the end-to-end performance of a particular data flow.

End-to-end measurements of wired IP networks has been an important area of research during last years. Therefore, until last years, the focus was only on wired networks and not on wireless networks.

End-to-end measurements of GPRS-EDGE networks have some particular problems that are not considered in the wired case. For example, the spectrum resource sharing adds some special issues, as well as the priority of the voice traffic.

The focus of this work is on end-to-end GPRS-EDGE networks capacity estimation and some other important performance issues. This work suggests a methodology for end-to-end measurements of this kind of networks. Based on this methodology a software tool was developed. Using that tool some experiments have been done to estimate different performance metrics in a GPRS-EDGE network.

The paper is organized as follows. A summary about the state of art concerning end-to-end measurements over IP networks is in Section 2. An explanation of important parameters to measure over GPRS-EDGE is done in Section 3. The measurements issues over GPRS-EDGE networks are presented in Section 4. An algorithm which deals with the matters presented in Section 4 is explained in Section 5. The implemented measurement system is explained in Section 6. Some results obtained with this system are presented in Section 7. In Section 8, the influence of the transport layer protocols on the throughput measurement is studied. Finally, the paper is concluded in Sections 9 and 10.

2. RELATED WORKS

The main research topics on end-to-end measurements on

the Internet are:

Link capacity or bottleneck link capacity estimation in a network path.

There are many suggested procedures in order to estimate the capacity of a wired link. Main techniques are based on sending fixed size probe packet pairs. At the receiver the inter-arrival time is analyzed and, knowing the packet size, the bottleneck capacity is estimated. Each technique fits better depending on the characteristics of the path and the cross traffic. In general, all techniques work if there is not cross traffic. However, in a heavy loaded network or in a path with cross traffic in many links, the errors in the estimation can be important [10]. Cross traffic is defined as packets which belongs to other connections which interferes with probe packets.

Internet tomography.

The goal of Internet tomography is to estimate link QoS parameters from end-to-end estimations of these parameters [1, 3, 5, 6, 11, 14, 18]. These works are related to the subject studied in this paper, because they contribute with methodologies and ideas to measure some end-to-end parameters. However, they do not solve the problems that arise in wireless networks.

Link or path "available bandwidth" measurement.

The available bandwidth (ABW) of a link i in the time interval $(t, t + \tau)$ is

$$A_i(t, t + \tau) = C_i(1 - u_i(t, t + \tau)) \tag{1}$$

where C_i is the link capacity and $u_i(t, t + \tau)$ is the average link utilization in the time interval $(t, t + \tau)$. The minimum A_i in the path is defined as the ABW of the path. There are many developed algorithms to estimate the available bandwidth. Pathload [7, 9, 8], PathChirp [15] and Spruce [17] are important examples of different techniques. Strauss et al. [17] compare Spruce with other tools used to estimate the ABW, like Pathload.

All these works are based on wired networks, where each link capacity is fixed and the packets share each link in a FIFO queue. This work builds a methodology that can be used for end-to-end performance estimation over cellular networks, where the previous assumptions are not necessarily true.

3. WIRELESS LINK PERFORMANCE MEASURES

This work is focused on the estimation of the following parameters.

- Instantaneous link capacity

- Mean link capacity

- Activity interval time

- Total throughput

- Mean throughput in an activity interval time

- Channel activity time

- Packet loss

- Round Trip Time (RTT)

- Jitter

These parameters will be estimated by sending probe packet pairs of constant size ϖ at constant rate. The inter-arrival time between pairs at the client is Δt. In the following subsections the meaning of these parameters is explained.

3.1 Instantaneous link capacity

In this work, the ratio between probe packet size and inter-arrival time between two probe packets that were queued together in the link is called *instantaneous link capacity*.

3.2 Mean link capacity

It is the sum of the *instantaneous link capacity* multiplied by the interval time in which occurs, over the sum of these time intervals.

$$\overline{C} = \frac{\sum_i C_i \Delta t_i}{\sum_i \Delta t_i} \tag{2}$$

3.3 Activity interval time

In a TDMA (Time Division Multiple Access) system, each client uses the channel during a certain period of time that, in this work, is called *activity interval time*.

3.4 Total throughput

Total throughput is defined as the amount of data transferred over the transference time.

3.5 Mean throughput in an activity interval time

The average throughput of all the activity interval times is called *mean throughput in an activity interval time*.

3.6 Channel activity time

Channel activity time is defined as:

$$Act = \frac{\sum T_{activity}}{T_{tot}} \tag{3}$$

where $T_{activity}$ represents a given activity interval time and T_{tot} is the total time of the experiment. This parameter is important because it shows the time percentage during which the client had the channel.

3.7 Packet loss

The packet loss is defined as:

$$Loss = 1 - \frac{P_{rc}}{P_{ss}} \tag{4}$$

where P_{rc} is the number of packets received by the client and P_{ss} is the number of packets sent by the server.

3.8 RTT

The time elapsed taken for an IP packet to travel to a remote place an back again is called *RTT*.

3.9 Jitter

The variation of succesive values of RTT is called *Jitter*.

4. MEASUREMENT METHODOLOGY

4.1 Measurement technique

In each experiment probe packets are sent at a constant rate for a short time period. In order to estimate the instantaneous bottleneck link capacity, probe packets must be enqueued together in the radio access buffer (assuming that the wireless link is the bottleneck). The packet rate to accomplish this is chosen as the maximum mobile capacity (that depends on the mobile class).

There is a tradeoff in the selection of the probe packet size. The accuracy in the analisys of each activity interval time is important. In order to have more samples in each activity interval time it is necessary to send packets with the smallest size allowed. However, in order to have less uncertainty in the capacity estimation from the measures of the inter-arrival times, it is preferred to use as big as possible packet sizes.

The time stamp absolute error is around 1 ms. To solve this tradeoff packet sizes around $300 - 500$ bytes are used. These sizes are the smallest that provide a capacity estimation relative error lower than 10% for the biggest GPRS-EDGE capacities that are around $120 - 240$ kbps.

Next subsection summarizes the most common used methodologies for wired link capacity estimation. In subsection 4.3 wireless link capacity estimation problems are explained.

4.2 Bottleneck link capacity estimation

For each packet pair the link capacity can be estimated as $\frac{\varpi}{\Delta t}$, where ϖ is the packets size and Δt is the inter-arrival time between consecutive packets at the client. After this estimation, the values obtained for each packet pair are classified for further processing. By this processing a final capacity is estimated.

If a pair of packets (with packet size ϖ) wait together in the bottleneck link buffer (with link capacity C), and in not any other link, the time between arrivals is $\frac{\varpi}{C}$. One important issue to point out is that packet pair estimation capacity may be distorted by cross traffic or by limitations on the measurement system.

There are two situations that may affect the measurements. The first situation happens if there is cross traffic between probe packets. In this case they will be separated a time T, which is larger than the separation imposed by the bottleneck capacity. In this case the system will estimate a fake bottleneck capacity $C' < C$. The other situation happens when the probe packets wait in queues after the bottleneck link. In this case they will join together yielding a fake bottleneck capacity $C' > C$. These effects are illustrated in figure 1. These cases must be filtered. The capacity estimation will be more accurate if the bottleneck link is not heavy loaded by cross traffic and the last queue of the path is at the bottleneck link.

Several techniques are used for processing packets pairs and estimate the capacity, filtering the cases explained before. Usually, histogram classification or kernel density estimation are used. Pathrate [4] and Nettimer [10] use these techniques.

The main idea of these algorithms is to estimate the capacity by finding the maximum of the distribution (or histogram). This value represents the estimated capacity that occurs more frequently.
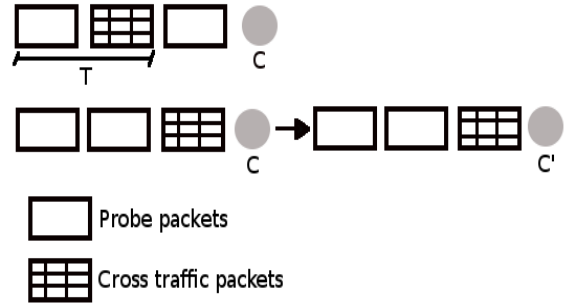

Figure 1: Cross traffic effect on bottleneck capacity measures.

Multiclass Slot	DL Slots	UL Slots	Active Slots
1	1	1	2
2	2	1	3
3	2	2	3
⋮	⋮	⋮	⋮
9	3	2	5
10	4	2	5

Table 1: Mobile classification by its multiclass slot type.

4.3 Wireless TDMA systems issues

4.3.1 Introduction

In general, in cellular networks, the bottleneck is located on the air interface. In addition to the problems explained before in the case wired link capacity estimation, the following issues arise in the case of cellular wireless links:

- Mobile class
- Carrier to interference ($\frac{c}{i}$) relation
- Available time slots for data transfer in the serving cell

Mobile class

The mobile class is a classification between different types of terminals which determines how many simultaneous time slots can be used by the client in order to transfer data. Table 1 lists most commonly used class types and the number of time slots used for download (DL) and upload (UL) in each class.

Carrier to interference relation

Carrier to interference relation $\left(\frac{c}{i}\right)$ is the parameter which determines the effective rate of data transfer per time slot. When $\frac{c}{i}$ relation is small, the error probability during the transfer is high. In order to avoid this problem, when $\frac{c}{i}$ relation is small, more redundancy bits are used. The more redundancy bits are used to transfer information, the less is the information rate per time slot. Table 2 list different available code schemes in EDGE (MCS).

Available time slots for data transfer

The service provider configures the number of available time slots for data transfer in the cell. So, some of the configu-

MCS	BW per TS (at layer 2, in kbps)
1	8.8
2	11.2
3	14.8
⋮	⋮
8	54.4
9	59.2

Table 2: Different code schemes in EDGE.
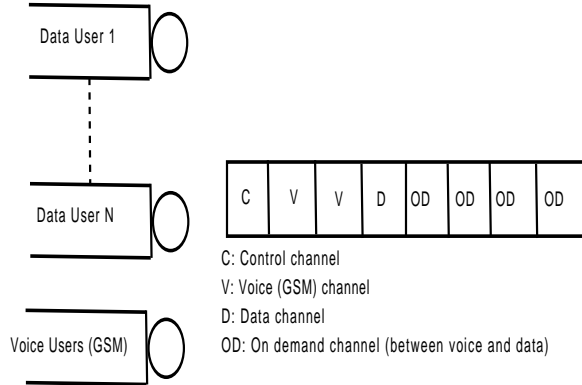
Figure 2: Typical GPRS-EDGE queueing system.

ration variables in the cell are the number of fixed time slots assigned to data transfer, the number of fixed time slots assigned to voice traffic and the number of on-demand time slots between voice and data. Usually, for the on-demand time slots, voice calls has preemptive priority over data transfers.

Cellular link capacity can not be estimated using the same procedures as in the wired links case because cellular links may have capacity time variations. This problem is analyzed in more detail in the following subsection.

4.3.2 GPRS-EDGE *Queueing System*

Figure 2 shows a typical GPRS-EDGE queueing system. As it can be seen, this system may include interaction between voice and data users. There are multiple factors that impact on the capacity estimation. In the following paragraphs different situations are analyzed in order of complexity.

Number of GSM *users are less or equal than fixed voice slots, there is only one data user in the cell and a fixed $\frac{c}{i}$ relation is assumed.*

In this case, voice users do not use on-demand time slots. All data time slots are available for the client. Therefore, the client has all data resources available all the time. The capacity will be only determined by the mobile class and $\frac{c}{i}$ relation. Assuming a fixed $\frac{c}{i}$ relation this case is similar to the classical wired one, like ADSL.

Number of GSM *users are higher than fixed voice slots, there is only one data user in the cell and a fixed $\frac{c}{i}$ relation is assumed.*

In this case there are more users than fixed voice slots assigned by the operator. So, voice users need to use on-

demand time slots, and they have priority over data users on these channels. In this case the data users can only use the fixed data time slots and the on-demand time slots that are not used for voice calls. So, the capacity is determined by the mobile class, $\frac{c}{i}$ relation and on-demand time slots availability (instantaneously determined by the amount of voice calls). In this case the link capacity is modulated by voice traffic.

Number of GSM *users are less or equal than fixed voice slots, the client shares the channel with other data users and a fixed $\frac{c}{i}$ relation is assumed.*

In this case the client is sharing data resources with other data users. Thus, each data user does not have the channel for his own use all the time. This case can be seen as a fixed capacity system (assuming a fixed number of users transmitting all the time with a fixed $\frac{c}{i}$ relation) shared in time with other data users.

Number of GSM *users are higher than fixed voice slots, the client shares the channel with other data users and a fixed $\frac{c}{i}$ relation is assumed.*

This is a more realistic case which can be seen as the superposition of the two cases mentioned before. So, in this case the link capacity is modulated by voice traffic and shared in time with other data users.

More complex cases.

A more realistic model is obtained by the superposition of all the above mentioned variables, which includes variation of data and voice traffic and $\frac{c}{i}$ relation. Another relevant effect is cross traffic inserted in the channel by the data client (another kind of traffic different from the probe packets that may introduce certain level of noise on it). The following algorithm deals with this situation.

5. ALCE: *AN ALGORITHM TO ESTIMATE CELLULAR LINK CAPACITY*

This algorithm is composed by different modules. The main modules are the following:

1. Activity and inactivity intervals classification

2. Analysis of each activity interval

   A. Potential capacities detection

   B. Process of samples for each one of the capacities detected in A

3. Parameters estimation

5.1 Activity and inactivity intervals classification

In order to find the activity intervals, an adaptation of the K-means algorithm [2] was developed. The algorithm calculates $K - 1$ thresholds and classifies the packet pair inter-arrival time into K groups, where K is supposed to be known. In our case $K = 2$ represents the group of samples that belong to one activity interval (group 1) and the group of samples that denotes a change of activity interval (group 2). Once the execution of the algorithm is finished, the selected threshold is given by
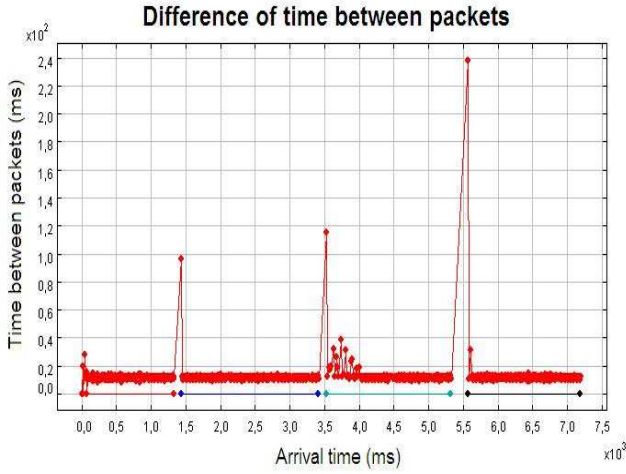
$$Th_{Kmeans} = \frac{\overline{g_1} + \overline{g_2}}{2} \qquad (5)$$

Figure 3: Experiment in which activity intervals are clearly denoted.

Figure 4: Experiment in which there is only one activity interval.

where $\overline{g_1}$ and $\overline{g_2}$ are the means of the groups 1 and 2 respectively.

One particular case is when during the experiment does not occur more than one activity interval. This happens when the client has the channel for his own use. In this case, at the moment of distinguishing between activity intervals, if we run the "pure" K-means algorithm, an incorrect discrimination is done, identifying some differences of time between consecutive packets as inactivity intervals.

In order to solve this problem a minimum threshold is calculated when detecting the intervals. The time between packet arrivals depends on the packet size and on the instantaneous capacity, so, a minimum threshold, Th_{min}, is calculated based on this parameters.

Applying this criterion, the final threshold applied to differentiate between inactivity periods is given by equation 6.

$$Thm = max\left\{Th_{min}, Th_{Kmeans}\right\} \qquad (6)$$

Figure 3 shows the situation of a common experiment, in which activity intervals are clearly denoted and separated by inactivity intervals. On the other hand, figure 4 shows an experiment in which there are not any inactivity intervals ,so, the adaptation of the K-means algorithm is necessary to obtain valid results.

As an example, if we run the "pure" K-means algorithm, figure 5 shows how a great number of false inactivity intervals are detected.

5.2 Activity interval analysis

After detecting activity intervals, the algorithm proceeds to analyze each one of them. The flow diagram of this module is shown in figure 6.

5.2.1 Potential capacities detection

Potential capacities of each activity interval are detected based on kernel density estimation technique [16, 19]. The main idea of this technique is the following. Given a finite sequence of n samples, $X_1, \ldots, X_n$, the algorithm estimates
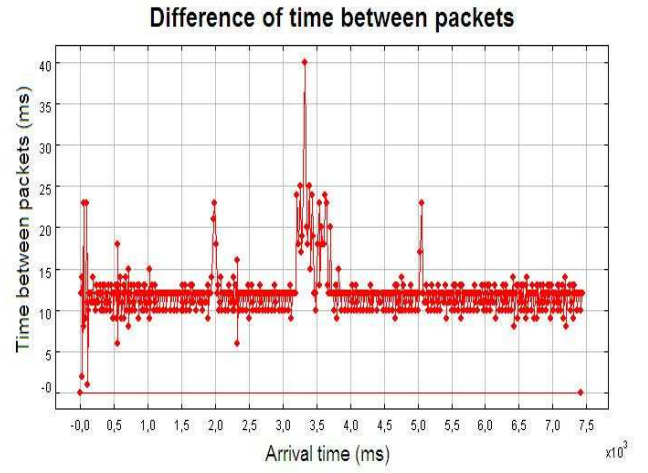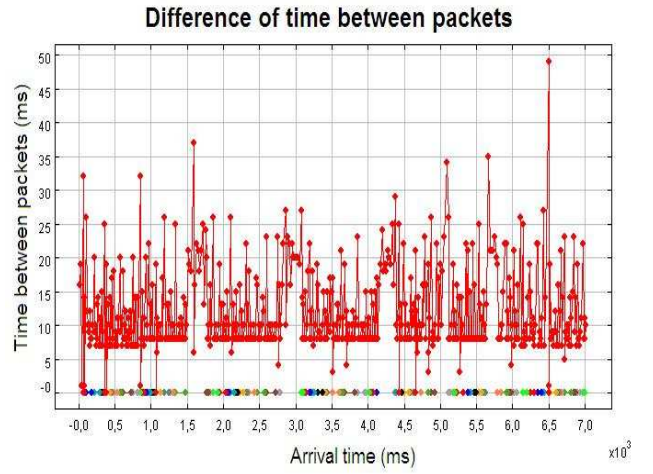

Figure 5: "Pure" K-means algorithm finds many fake inactivity intervals.

the density function $f(t)$ as

$$\hat{f}(t) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{t - X_i}{h}\right) \qquad (7)$$

where K is a kernel function and h is the bandwidth. The kernel has the following properties

$$\int_{-\infty}^{\infty} K(t)dt = 1, K(t) \geq 0 \qquad (8)$$

The Kernel used in the algorithm is the Epanechnikov Kernel, defined as

$$K(t) = \frac{3}{4}(1 - t^2)I_{|t|<1} \qquad (9)$$

where

$$I_{|t|<1} = \begin{cases} 1 & \text{if } |t| < 1 \\ 0 & \text{in other case} \end{cases}$$

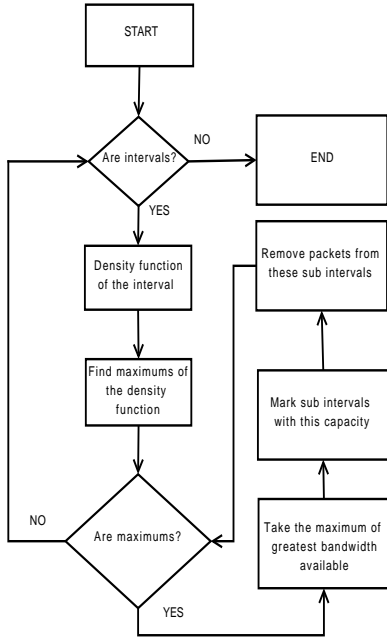The optimum Epanechnikov kernel bandwidth h is given

Figure 6: **Flow diagram of the activity interval analysis.**

by the following equation

$$h_{opt} = (40)^{1/5}\pi^{1/10}s_n n^{-1/5} \tag{10}$$

where s_n represents the standard deviation of the samples.

$$s_n = \left(\frac{1}{n}\sum_{i=1}^{n}(X_i - \overline{X})^2\right)^{1/2} \tag{11}$$

After the density function is obtained the algorithm proceeds to find its maximums, being these points the potential capacities during the activity interval.

5.2.2 *Samples processing to avoid false potential capacities detected*

In this module, the algorithm evaluates if these potential capacities really correspond to an activity interval.

In order to discard false capacities a set of rules are applied to the potential capacities. For example, rules to avoid capacities that are not possible in a cellular wireless link, rules to avoid spurious measurements or rules to take into account capacity variations due to the limited accuracy of the measurement procedure can be considered.

At the end of this procedure all valid capacities in the activity interval of study are saved. The process is repeated with the next activity interval until there are no more activity intervals to analyze. When this module is finished, the algorithm executes the following one.

Figure 7 shows a common situation in the analysis of an activity interval. In this case, a great quantity of samples register the same capacity. However, the presence of certain isolated samples which are far from this value could induce to think, at first sight, in the presence of many capacities during the activity interval. However, due to the rules imposed to filter spurious capacities, the algorithm identifies these isolated points as invalid measurements done by the
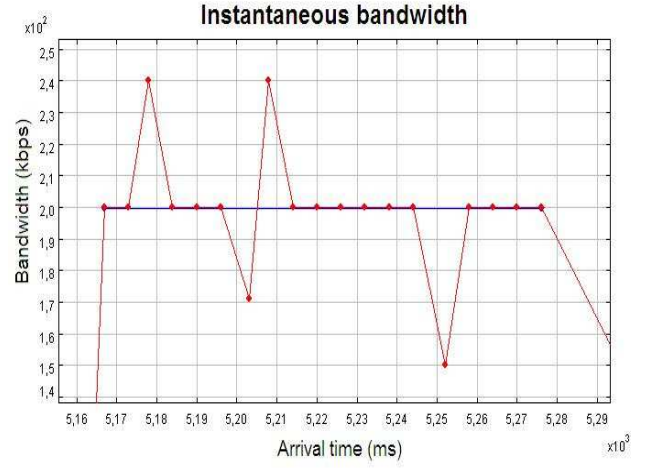

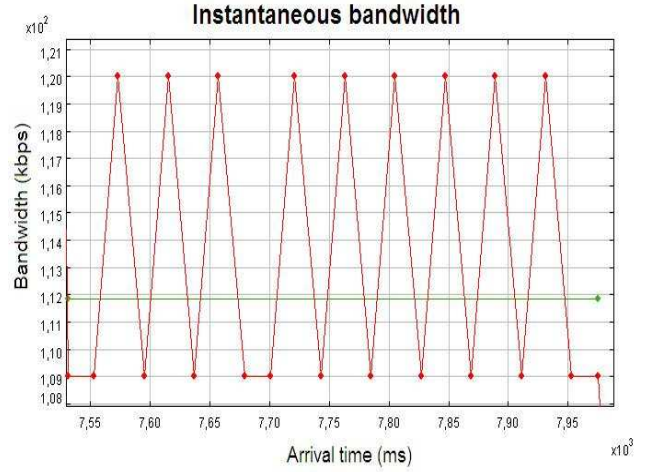Figure 7: **Capacity detection in presence of low level of noise.**

Figure 8: **Capacity variations due to time stamps accuracy.**

system or measures affected by cross traffic. For these reasons the algorithm behaves as shown in figure 7.

In figure 8 a particular case is shown. In this experiment the instantaneous link capacity oscillates between 120 and 109 kbps. This situation is probably caused by the accuracy in the time stamps. Particularly, in this case, the experiment was done with a packet size of 300 bytes, while the registered inter-arrival time were between 22 and 20 ms. However, it is reasonable to suppose that the packets arrived at the client at a rate between 22 and 20 ms. This means that the real capacity is between 120 and 109 kbps. It can be seen how, by the application of the kernel density estimation, the final capacity is detected at 113 kbps.

Figure 9 shows a case in which the experiment was done under several noise conditions. As it can be seen it is not easy to find the real link capacity. Under these conditions the capacity estimation would have large uncertainty, so the algorithm does not mark any capacity.
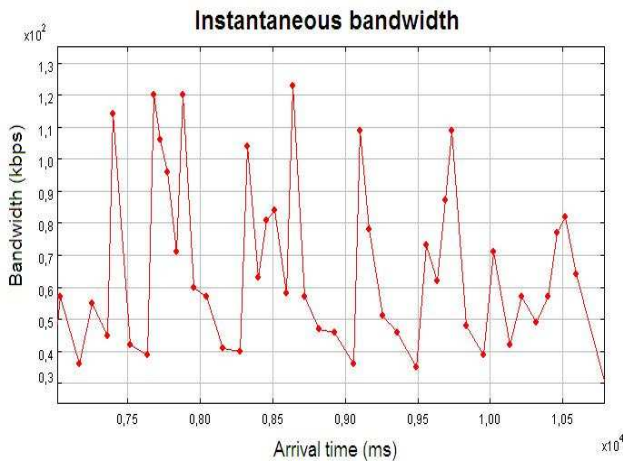
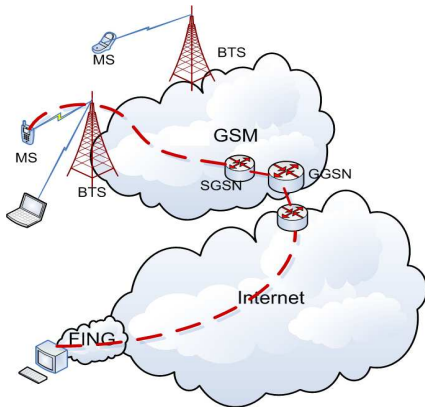**Figure 9: Capacity detection with high level of noise. The algorithm is not able to distinguish any capacity.**

Figure 10: Network measurement system.

6. IMPLEMENTED MEASUREMENT SYSTEM

The implemented measurement system consists of a server running Linux and a Servlet. The client can use an Applet (if the client is using a GPRS-EDGE modem) or a Midlet (if the client is using a cellular phone only). Figure 10 shows a diagram of the measurement system. The system architecture is explained in more detail in [12].

The measures are done in two phases:

- Phase I: Edge router deceive
- Phase II: Experiment

6.1 Phase I: *Edge router deceive*

The local GPRS-EDGE provider assigns IP addresses to its clients from a private pool through an edge router applying NAT[1]. For this reason, it is necessary to find a way to establish the communication between the Internet server and the client in order to make the corresponding measurements. This first phase "deceives" the edge router in order to establish a path from the server to the client.

---
[1]Network Address Translation

At the beginning of the experiment, its parameters are negotiated between the client (Applet or Midlet) and the server via HTTP packets. After this negotiation, the server begins the experiment by sending UDP probe packets with the chosen parameters. Since the initial request for the experiment is done by the client, the edge router register an exchange of HTTP packets from the client (which has a private IP address) to the server (with a public IP address). Once this exchange is finished the server starts sending UDP packets to the client. If the server sends the UDP probe packets to the private client's IP, the packets will be discarded by the router. If the server sends the probe packets to the edge router public IP, the packets will be discarded, because the router does not know which client has requested these packets.

To solve this problem, the system implements the "deceive" phase. In this phase the client sends false probe packets to the server immediately after the HTTP negotiation. By doing this, a register on the NAT table of the edge router associates an UDP flow from the client to the server in any given pair of ports. The server reads the information of this UDP packets and starts the experiment by sending the probe UDP packets to the client using the ports chosen by the client in the deceiving phase. In this case, when the experiment packets arrive at the edge router, a record on the NAT table exists, so the router knows that this flow belongs to the client - server path and routes the packets correctly.

6.2 Phase II: *Experiment*

The experiment sends UDP packets from the server to the client in order to gather the needed information to run the previous algorithm. The packets sent from the server, are rebounded at the client and received by the server. This packets contain useful information for the algorithm as a sequence number and the arrival time stamps at each end of the path.

7. RESULTS

During the trial period, many experiments were done under different circumstances. The variables taken into account were the *place of the experiment* (cell of service), $\frac{c}{i}$ *relation*, *hour of the day* and the *mobile type*. Experiments were done with two mobiles, Mobile A and Mobile B. Both were used as modems, connected to a PC and using the Applet as the client software or as a mobile client, using the Midlet as the client software. The places chosen to run the experiments were different neighborhoods of Montevideo (Punta Carretas, Malvín, La Blanqueada and Palermo) and a seaside (Playa Hermosa).

7.1 Mean capacity

Table 3 shows maximum, mean and minimum values of capacities registered during the experiments. Punta Carretas, Palermo and Playa Hermosa (P.H.) were the chosen places, and the measures were done using Mobile A.

In these experiments was observed that under normal traffic conditions mean capacities were between 190 and 220 kbps. These results reveal that Mobile A obtains, in general, four time slots for download with code scheme between MCS-6 and MCS-9.

Figure 11 shows experiments made in Playa Hermosa. In this case the pattern of mean capacities on the first three days of experiments done in the afternoon were greater than

Place of study	Mean capacity (kbps)	Min capacity (kbps)	Max capacity (kbps)
P. Carretas	200	167	212
Palermo	202	187	224
P.H. - Afternoon	196	142	216
P.H. - Night	142	115	203

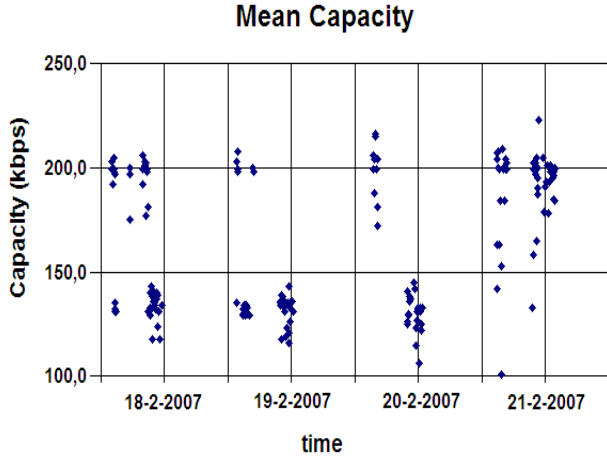Table 3: Mean capacities registered by place with Mobile A.

Figure 11: Mean capacity in kbps in Playa Hermosa's cell with Mobile A.

the mean capacities registered at night. This is because the experiments were done during a holiday weekend in a seaside, and at night the voice traffic is more important. Also, it can be seen that this phenomenon is not appreciated the last day of the holiday, because the majority of the people had returned to their residence cities.

Figure 12 shows the capacity obtained with both mobiles in Malvín. As it can be seen, mean capacities registered with mobile B are much smaller than the ones registered with Mobile A. Doing several experiments in different hours and places, the influence of factors like cell of service or the time of the experiment were discarded; then such behavior was caused by the type of mobile used.

7.2 Channel Activity time

As mentioned in Section 3, channel activity time is an indicator of the amount of time that the client is using the resources. Figure 13 shows the channel activity time registered with both mobiles. As it can be seen, channel activity time of mobile B was bigger than channel activity time of mobile A.

7.3 RTT

After the experiment phase was finished, we concluded that the delays registered on the network were high in order to provide real time applications.

Values obtained with both mobiles were different, as it can be seen in table 4. Cell of service also have influence over the RTT obtained, because of the $\frac{c}{i}$ relation or the amount
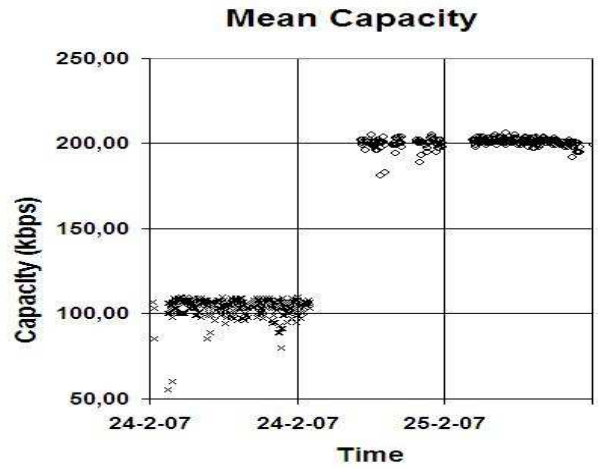

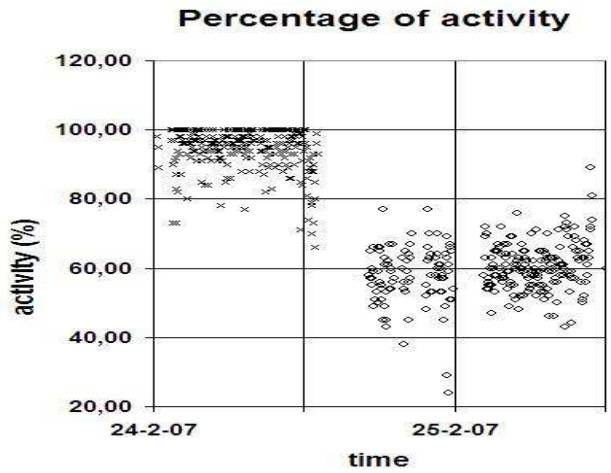Figure 12: Mean capacity with mobile B (cross) and A (circles).

Figure 13: Mobile B (cross) and Mobile A channel (circle) activity time.

of traffic on it, as table 5 shows.

7.4 Jitter

From the experiments done we conclude, as expected, that registered jitter is tightly related to RTT.

Table 6 shows values obtained in different places. This range of values entail that certain services which require low jitter values, like Push To Talk, suffer a high degradation of performance.

7.5 Packets loss

The registered loss in the majority of cases was between 0 and 4 % of packets sent. However, in some experiments this ratio was above 10 %.

Mobile used	Mean RTT (ms)	Min RTT (ms)	Max RTT (ms)
Mobile A	1329	833	2318
Mobile B	897	709	1803

Table 4: Registered RTT in Punta Carretas cell with both mobiles.

Place of study	Mean RTT (ms)	Min RTT (ms)	Max RTT (ms)
P. Hermosa	2376	920	8043
P. Carretas	1329	833	2318

Table 5: RTT obtained in different places with mobile A.

Place of study	Mean Jitter (ms)	Min Jitter (ms)	Max Jitter (ms)
P. Hermosa	36	22	92
P. Carretas	23	5	57

Table 6: Jitter obtained in different places with mobile A.

Figure 14: Measurement topology.

8. THROUGHPUT VS CAPACITY: INFLUENCE OF THE TRANSPORT LAYER PROTOCOL AND THE ROUND TRIP TIME

The previous section analyzes the cellular link capacity and other important performance parameters. The focus of this section is on the throughput. The throughput is an important parameter at the application layer. Even if the system has high capacity, such capacity can not be used unless a high throughput is acheieved.

Link *throughput* is an estimator of data transfer rate that an application can achieve, whereas *capacity* refers to the real "physical" capacity of the access medium. Taking these considerations into account, the maximum throughput that an user could achieve is equal to the link capacity.

There are many parameters that influence the throughput. The capacity of the bottleneck link is an important parameter, but the RTT and the transport layer protocol have an important influence too.

In our experiments the throughput obtained by a GPRS-EDGE connection is in many cases, more than two times lower than the bottleneck link capacity. The problem to analyze is why there is such a big difference. With this purpose, several measures have been done directly on the net, using different kinds of traffic between a server and a mobile client. Figure 14 shows the topology used on the experiments. The client has a GPRS-EDGE modem, while in the other side of the communication resides an FTP server, from which different file transfers and measurements have been done.

All measurements have been done late at night. This period was chosen in order to assume a low charge of the cellular network. $\frac{c}{i}$ relation was high enough to ensure that during all the experiments use EDGE channels with high MCS. With these assumptions, it is supposed that the client will achieve high transfer rates.

8.1 Study of throughput for TCP traffic in the downlink

In this experiment multiple large files were downloaded, studying achieved throughput by the user (figure 15).
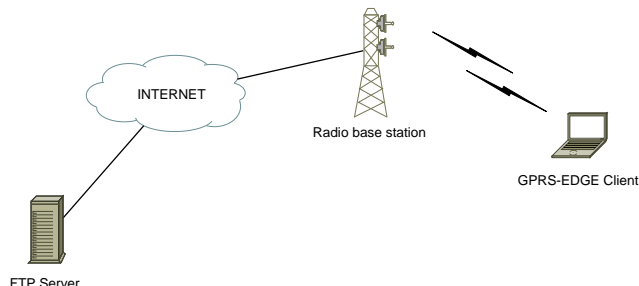
To do such measurements we have used a traffic snif-fer on users terminal (Ethereal) and a software that monitors the rate of bits that arrives to the modem (NetStat Live). At first, one download was done, studying stationary achieved throughput. Then the number of simultaneous downloads was increased, studying again the aggregated stationary throughput. This procedure was continued until aggregated throughput stop rising.

8.2 Study of capacity and throughput in uplink for different types of traffic

In this experiment a tool for capacity estimation was used, Iperf. This tool is based on TCP and UDP traffic in order to measure real uplink capacity and achieved uplink throughput for the client for different kinds of traffic.

Estimations obtained with this tool depend on the kind of traffic which is used. In the case that this traffic is UDP, estimation consists in sending one or more bursts of UDP flows from the client to the server, flooding the link and analyzing the packets in the server, as shown in figure 16. In case that this traffic is TCP, one or more of TCP connections are opened, determining aggregated throughput at the server, as shown in figure 17.

8.3 Results

8.3.1 Study of throughput for TCP traffic in the downlink

The number of FTP transferences done, in chronological order, was: 1 file, 2 files, 3 files, 5 files, 1 file, 4 files and 5 files. Figure 17 shows the achieved throughput for each case.

Table 7 shows achieved throughput in the experiment varying the number of FTP transferences.

This experiment shows how the achieved throughput from a single TCP connection did not reach channel capacity. The link capacity is around 180 kbps, because the mobile was using 3 slots for download (3+2 configuration). This case shows that one connection can achieve only a third of
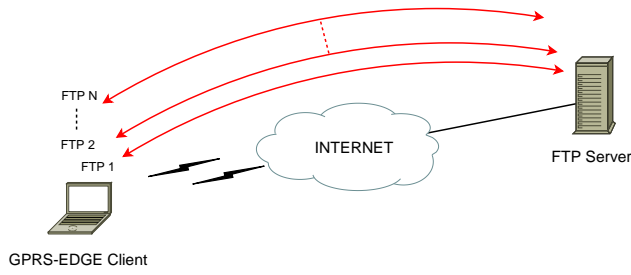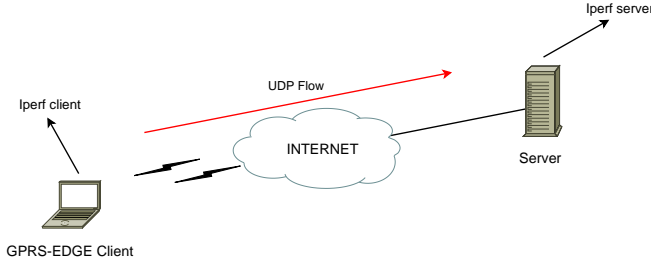
Figure 15: Downlink TCP throughput.

Figure 16: Uplink throughput.

Numb. of FTP transferences	Throughput (kbps)
1	58,3
2	96,1
3	135,6
5	161,5
1	64,8
4	146,2
5	158,5

Table 7: Throughput estimation through simultaneous FTP downloads.

Number of TCP connections	Estimation (kbps)
1	56,3
2	74,2
3	91,5

Table 8: Estimation of uplink capacity with Iperf using TCP connections.

Number of UDP flows flows	Estimation (kbps)
1	114
2	112
3	117

Table 9: Estimation of uplink capacity with Iperf using UDP flows.

the link capacity. However, this value could be increased by opening some simultaneous connections. For five simultaneous connections achieved throughput is close to the link capacity.

It is known that TCP throughput cannot be equal to the link capacity because its congestion control algorithm decreases the throughput. However, in this case the difference seems to be important. The problem is that TCP throughput is inversely proportional to the RTT [13]. As it can be seen in Table 5, in all cases the RTT is high compared to usual values obtained in typical wired links. The RTT is normally high in cellular networks and this value generates the small throughput that can be obtained by one connection.

8.3.2 Study of capacity and throughput in uplink for different types of traffic

Tables 8 and 9 summarizes estimations for both kinds of traffic and different number of connections/flows.

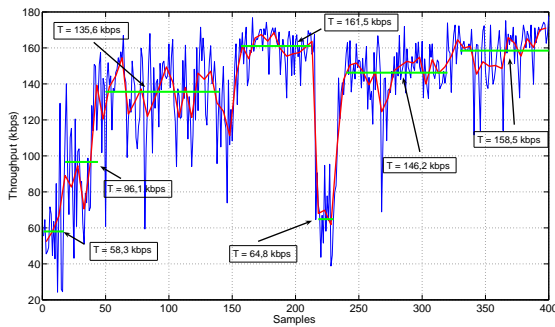To analyze the results, it is important to consider the characteristics of the different kinds of traffic. In the case of UDP traffic, estimations are equal to the uplink capacity, independently of the number of flows. In the case of TCP traffic, all the results obtained were the same as in downlink, only one connection achieved a lower throughput than a few simultaneous connections.

In this sense, capacity estimation using TCP traffic is not recommended. For this reason, it was implemented in the system end-to-end measurements using UDP packets.

9. CONCLUSIONS

In this work we have introduced both, an end-to-end measurement system and a specific algorithm to measure performance parameters in TDMA based IP networks, particularly on GPRS-EDGE. This algorithm has the particularity of being specifically designed for the analysis of the GPRS-EDGE protocol, obtaining big amounts of information for it.

Several experiments have been done in order to test the robustness of the system and the algorithm, obtaining good

Figure 17: Downlink throughput during the experiment.

results. Due to this results we have done an extensive measurement on a GPRS-EDGE network, getting to know its main characteristics.

The developed system can be used through any GPRS-EDGE network. The only requirement is to have a Java Virtual Machine installed on the client (in case that the client is connected to the GPRS-EDGE network through a modem) or a Java capable mobile.

10. ACKNOWLEDGMENTS

11. REFERENCES

[1] M. Adler, T. Bu, R. Sitaraman, and D. Towsley. Tree layout for internal network characterizations in multicast networks. *NGC '01: Proceedings of the Third International COST264 Workshop on Networked Group Communication*, pages 189–204, 2001.

[2] F. Bon. Técnicas no supervisadas. Métodos de agrupamiento. course notes from universidad de granada. 2001.

[3] R. Cáceres, N. G. Duffield, Horowitz, and D. Towsley. Multicast-based inference of networkinternal loss characteristics. *IEEE Transactions on Information Theory Vol. 45*, pages 2462–2480, 1999.

[4] C. Dovrolis, P. Ramanathan, and D. Moore. What do packet disperssion techniques measure? *Infocom*, pages 905–914, 2001.

[5] A. Downey. Using pathchar to estimate internet link characteristics. *Measurement and Modeling of Computer Systems*, pages 222–223, 1999.

[6] V. Jacobson. Pathchar, a tool to infer characteristics of internet path. Presented at the mathematical sciences research institute. 1997.

[7] M. Jain and C. Dovrolis. Pathload: A measurement tool for end-to-end available bandwidth. *Proceedings of Passive and Active Measurements (PAM)*, pages 14–25, 2002.

[8] M. Jain and C. Dovrolis. Pathload: A measurement tool for end-to-end available bandwidth. *In Proceedings of Passive and Active Measurements (PAM) Workshop*, 2002.

[9] M. Jain and C. Dovrolis. End-to-end available bandwidth: measurement methodology, dynamics, and relation with TCP throughput. *IEEE/ACM Transactions in Networking*, 2003.

[10] K. Lai and M. Baker. Nettimer: A tool for measuring bottleneck link bandwidth. *Proceedings of the USENIX Symposium on Internet Technologies and Systems*, 2001.

[11] E. Lawrence, G. Michailidis, and V. N. Nair. Inference of network delay distributions using the EM algorithm. *Technical Report, University of Michigan*, 2003.

[12] J. A. Negreira, J. Pereira, and S. Pérez. MetroCel: Estimación de performance sobre enlaces GPRS-EDGE, 2007. http://iie.fing.edu.uy/~javierp/MetroCel.

[13] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose. Modeling TCP throughput: A simple model and its empirical validation. *Proceedings of the ACM SIGCOMM '98 conference on applications, technologies, architectures, and protocols for computer communication*, pages 303–314, 1998.

[14] F. L. Presti, N. G. Duffield, J. Horowitz, and D. Towsley. Multicast-based inference of network-internal delay distributions. *ACM/IEEE Transactions on Networking Vol 10*, pages 761–775, 2002.

[15] V. Ribeiro, R. Riedi, R. Baraniuk, J. N. J., and L. Cotrell. PathChirp: Efficient available bandwidth estimation for network paths. *Passive and Active Measurement Workshop*, 2003.

[16] B. Silverman. *Density estimation for stadistics and data analysis.* 1986.

[17] J. Strauss, D. Katabi, and F. Kaashoek. A measurement study of available bandwidth estimation tools. *Internet Measurement Workshop, Proceedings of the 2003 ACM SIGCOMM conference on Internet measurement*, pages 39–44, 2003.

[18] Y. Tsang, M. Coates, and R. Nowak. Network delay tomography. *IEEE Transactions on Signal Processing. Vol. 51*, pages 2125–2136, 2003.

[19] B. A. Turlach. Bandwidth selection in kernel density estimation. *A review, Discussion Paper 9317, Institut de Statistique, Voie du Roman Pays 34, B-1348 Louvain-la-Neuve, Belgium*, 1993.