Influence of the Packet Size on the One-Way Delay in 3G Networks

Patrik Arlos and Markus Fiedler

Blekinge Institute of Technology
Karlskrona, Sweden
{patrik.arlos,markus.fiedler}@bth.se

Abstract. We currently observe a rising interest in mobile broadband, which users expect to perform in a similar way as its fixed counterpart. On the other hand, the capacity allocation process on mobile access links is far less transparent to the user; still, its properties need to be known in order to minimize the impact of the network on application performance. This paper investigates the impact of the packet size on the minimal one-way delay for the uplink in third-generation mobile networks. For interactive and real-time applications such as VoIP, one-way delays are of major importance for user perception; however, they are challenging to measure due to their sensitivity to clock synchronisation. Therefore, the paper applies a robust and innovative method to assure the quality of these measurements. Results from measurements from several Swedish mobile operators show that applications can gain significantly in terms of one-way delay from choosing optimal packet sizes. We show that, in certain cases, an increased packet size can improve the one-way delay performance at best by several hundred milliseconds.

1 Introduction

Increasingly many devices use mobile connectivity for the exchange of data. Users expect the emerging mobile broadband to perform in a similar way as its fixed counterpart, no matter to which extent the medium is shared. In the third generation of mobile communications, represented by WCDMA and HSDPA, the per-user capacity allocation depends amongst others on the radio conditions, the user density, the mobility pattern, the offered traffic, etc. It is, however, not communicated explicitly towards user and the applications that might need this information for yielding the best performance, given the specific allocation. This imposes the need for end-to-end measurements with the goal to highlight network impact on the performance parameters of interest.

Given this background, this paper investigates the impact of the packet size on the minimal one-way delay (OWD) for the uplink in third-generation mobile networks, which is an important performance parameter for interactive and real-time applications. In particular, the minimal OWD provides information about the best-possible performance with given settings, undisturbed by congestion, radio problems, etc. However, we should not omit importance for the uplink's

OWD behaviour as it will affect the TCP performance accross mobile networks, as the acknowledgement packets utilizes this link. However, in this paper, we will not investigate the average and maximum OWD values for the uplink, as these are more likely to exhibit temporal and spactial artifacts, varying from test to test. As OWD measurements are very sensitive to clock synchronisation issues, the paper also presents and demonstrates a robust and innovative method to assure the quality of these measurements. We focus on the minimal OWD as our goal is to investigate the best-possible system behaviour, instead of the statistical behaviour of the system. Hence, our results can be seen as best case with regards to the OWD perceived by an application.

The paper is organized as follows. First, we describe and verify the measurement method in Section 2. Section 3 describes and discusses the experimental setup and analysis procedure. In Section 4, minimal OWD in several Swedish networks are evaluated as a function of the packet size. Section 5 concludes the paper and points out future work.

2 Method

The fundamental problem when evaluating the OWD is how to handle the clock synchronization. The OWD is, as such, simple to calculate. The OWD of the i^{th} packet, d_i, is calculated as:

$$d_i = T_{i,b} - T_{i,a} \tag{1}$$

where $T_{i,a}$ is the arrival time of the i^{th} packet at location a; correspondingly $T_{i,b}$ is the arrival time of the same packet at location b. In the general case, the time stamps ($T_{i,x}$) are obtained from two different clocks. To get an unbiased OWD estimate, these clocks should be synchronized. In [1] the authors investigate the three main synchronisation methods NTP, GPS and IEEE1588 used for OWD measurements. Usually the Network Time Protocol (NTP) [2] is used. This enables the clocks to be synchronized within $10 - 20$ ms for WAN, and < 1 ms for LAN. If the synchronisation needs to be better, then a GPS solution is needed. Together with NTP, this allows a synchronisation in the order of 1 μs. The current state of the art is to use Endace [3] DAG cards together with a GPS, then the theoretical synchronisation is in the range of 60 ns. However, according to our own experience [6], this is difficult to obtain in practice, as we still have two independent clocks. In [4] the author described the internal functioning of the time-keeping in side of the DAG cards, and in [5] the authors describe a method to synchronize clocks accross the Internet. Regardless of what method or technique used for synchronisation, the OWD estimations can at the worst be twice that of the synchronisation level [6].

Our method uses wiretaps and a special wiring in conjunction with DAG cards to obtain the time stamps from the same clock. In Figure 1 a schematic of the wiring is shown. When a packet is sent from SRC to DST it will travel across the upper wire (dashed line). As is passes the first wiretap A, a copy of the packet is made and is sent to the interface dag00, where it arrives it at time $T_{1,\text{A}}$. At the same time the original packet makes its way across the network and

eventually reaches wiretap B. Here, a copy is sent to interface dag01, where it is received at $T_{1,\mathrm{B}}$. Similarly, if a packet is sent from DST to SRC the packets are duplicated by the wiretaps and made available to the dag1x interfaces. The main drawback with this wiring is that we require close proximity between SRC and DST. The actual distance is determined by the technology that carries the traffic from the wiretap to the DAG cards. The main benefit with this wiring, is that the packet will be time stamped by the same clock, thus subject to the same drift/skew if present.

Let t_0 be the time when the packet actually passes wiretap A, and t_1 when it passes wiretap B. Then $T_{1,\mathrm{A}} = t_0 + L_a/P_s$, where L_a is the cable length between wiretap A and dag0, and P_s is the propagation speed in that cable. Similarly, we define $T_{1,\mathrm{B}} = t_1 + L_b/P_s = t_0 + L/P_s + L_b/P_s$, where L_b is the cable distance from wiretap B to dag0, and L is the cable distance between wiretap A and B. The OWD is then obtained as:$\Delta = T_{1,\mathrm{B}} - T_{1,\mathrm{A}} = L/P_s + \frac{L_b - L_a}{P_s}$. So if we select $L_b = L_a$ we cancel the second factor and obtain the desired OWD between the wiretaps.

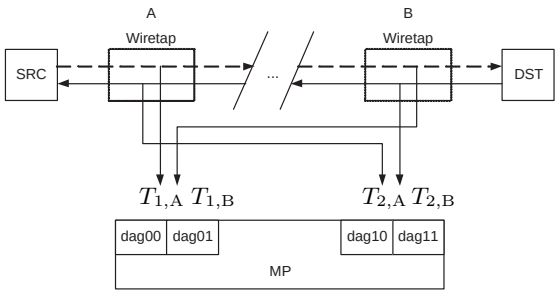

Fig. 1. Wiring method

To verify the method, we conducted two experiments using the setup as shown in Figure 1. In the first experiment the network in between the wiretaps was replaced by a 10 m CAT5e cable, in the second a 25 m cable was used instead. The CAT5e cable has a P_s of $0.59c \sim 0.64c$ [7, 8], where $c = 299\,792\,458$ m/s. The theoretical propagation time (P_t) is then between 52.1 ns and 56.6 ns for the first experiment, and 130.3 ns and 141.4 ns for the second. In Table 1 we show the corresponding results. As we are limited by the DAG 3.6 card resolution of approximately 60 ns [4, 6], our OWD values are multiples of this. Our results are within the span given by the theoretical results. Obviously, the method allows us to accurately detect changes in the OWD on the scale of a few nanoseconds.

Exp	Minimum [ns]	Mean [ns]	Max [ns]	Std.dev [ns]	Theoretical [ns]
1	0.0	54.7	119.3	17.4	$52.1 \sim 56.6$
2	119.0	139.2	179.0	28.2	$130.3 \sim 141.4$

Table 1. Summary statistics from the verification experiments.

3 Setup

To evaluate the mobile networks, we used the setup shown in Figure 2. Here SRC is sending traffic to DST. This is done via a Gateway (GW) that uses a Huawei E220 USB modem to connect to the mobile network. In-between the SRC and GW, we placed wiretap A. The other wiretap B is placed just in front of DST. The SRC and DST (both are P2-400 MHz with Linux 2.4 kernels) are connected with 10 Mbps full-duplex Ethernet cards (3Com). The GW is a Dual AMD Athlon 64 with 2 Gbytes of RAM (Windows XP SP2). The GW was configured for Internet sharing of the mobile network and no firewall was active. The SRC computer connected directly to the built-in Ethernet card (Broadcom) of the GW. The wiretaps feed into a Distributed Passive Measurement Infrastructure [9] enabled Measurement Point (MP) that stored the packet trace to file. Furthermore, the DAG cards were synchronised using both NTP and GPS.

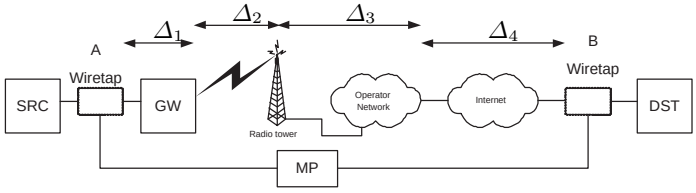

Fig. 2. Setup used in experiments.

3.1 Traffic Generation

To generate data we used a C++ program that sends UDP datagrams and allows us to control packet sending rate and datagram size. Furthermore, the progam uses an application layer header with three fields. These fields allow us to separate experiments (experiment id), experiment run (run id), as well as packets within a particular experiment run (sequence number). The sequence number starts at zero and is incremented by one for each transmitted datagram. Based on these three fields, we can uniquely identify each packet, thus avoiding any ambiguities associated with hashing.

During the evaluation of the mobile networks we used two streams running in parallel. The first sends one packet of size K bytes every T_s second, which

is done 200 times. It then waits for a fixed amount of time, and then starts to send another batch of 200 packets, this time with a larger packet size. The procedure is then repeated for all the packet sizes we wish to investigate. The second stream runs continously throughout the evaluation of all the different packet sizes, sending one 48 byte packet every 10 second. The purpose with this stream is to detect any time-of-day based variations in the network. As the purpose is to find the OWD, we do not want to stress the system so that it needs to queue our traffic. Using the two streams we will at most inject $(1468 + 48)$ bytes during one second.

3.2 Delay Calculation

As we are in control of both sender and receiver, we can easily identify both sending and receiving IP address as well as UDP port numbers. We then use the application header for the individual packet identification. Once identified, we can calculate the OWD for the individual packets. Using the same notation as before, the delay would be calculated as defined in Equation 1. However, due to numerical issues [6], this is not recommended. It is better to use the following equation:

$$d_i = \widetilde{T_{i,a}} - \widetilde{T_{i,b}} \quad \widetilde{T_{i,x}} = T_{i,x} - \lfloor T_1 \rfloor, \tag{2}$$

where T_1 is the arrival time of the first packet leaving the sender in that experiment. This will avoid having the time stamps truncated by the precision of the analysis tool.

3.3 Delay Components

The OWD that we will calculate has four contributors, see Figure 2. Δ_1 is the delay contribution by the GW, Δ_2 that of the radio network, Δ_3 that of the core network of the operator, and the last contribution Δ_4 comes from the Internet. Out of these four, we cannot measure or estimate Δ_2 and Δ_3 alone, as this means entering the domain of the operator.

We can estimate Δ_4 by using ICMP ping to the operator's Internet exchange. From our vantage point in the Internet, the operators are between five or six hops away, and between us and them we have the Swedish University Network (SUNET) [11] with optical multi-gigabit links. Hence the impact of this will not be negligble, but it will be quite small and stable, the average RTT between DST and the operator Internet exchange is 15 ms for all three operators. Hence, as the links are symetrical, the OWD contribution will be around 7.5 ms. Furthermore, we can ignore the packet size as the links have such high capacity that the serialisation delay is neglible [10].

In order to quantify Δ_1, we designed a special experiment. Instead of using the E220 USB, we replaced it with a D-link DUB-E100 FastEthernet USB adaptor that allows us to connect directly to the destination through Ethernet. As we are using a USB NIC, the packets travelling across this NIC will receive the same treatment as those that are sent across the modem.

From the collected data, the second stream did not detect any time-varying behavior. In fact, 37% of the packets experienced a delay less than 0.11 ms, and the maximum delay was 1.2 ms. The mean was 0.4855 ms with a standard deviation of 0.3362 ms. Apart from this, the GW seems quite stable in its handling of the packets. In Figure 3 we show the OWD through the GW for different packet sizes. We see that the OWD increases linearly, as expected [10]. The peaks noticable for the maximum values, are the result of single packets experiencing larger delays. Furthermore, the largest minimum OWD is just above 1.3 ms. If we use the minimum delay as a base, we can construct a rough model for Δ_1 given in ms:

$$\Delta_1(L) = 8.354\text{e}{-5} \cdot L + 0.078 \tag{3}$$

Here L represents the IP packet length in bytes. The 0.078 ms represents the minimal time through the GW, and the constant (8.354e−5) corresponds roughly to the capacity of the interface, i.e. 10 Mbps.

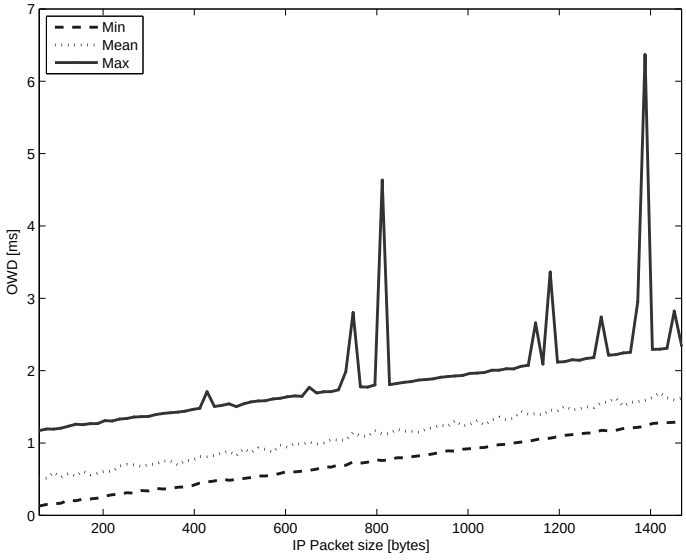

Fig. 3. Minimal OWD through the GW for different packet sizes.

As both Δ_1 and Δ_4 turned out to be significantly smaller than the OWD measured in the subsequent experiments, they are neglected from now on.

4 Evaluation of Mobile Networks

We conducted experiments on three different Swedish operators, the experiments were conducted at the end of September and early October 2009. Two of them (A

and B) share the radio access (RA), while the third (C) uses a different RA. The experiments focus on the OWD of the uplink; furthermore, all experiments were done while the sender was stationary. We focus on the uplink as this is believed to be the narrowest part in the end-to-end chain. Furthermore, to minimize the impact of daily patterns, congestion, radio problems, queueing, etc., we focus our analysis on the minimum OWD obtained in the experiments. However, just to give an indication on the environments, mean OWD was in the range of 140 ms to 700 ms, while the maximum was between 200 and 3870 ms, and the standard deviation was found between 7 ms and 480 ms.

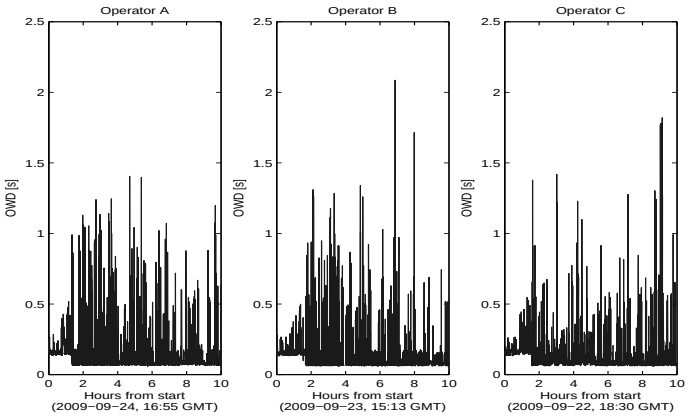

Fig. 4. Long term evaluation of the operators.

In the first experiment we investigated IP packet sizes starting at 60 bytes, and incrementing 16 bytes for each run, up to 1468 bytes, at a sending rate of one packet per second. We start by looking on the variations over time in the OWD of individual packets (one 48 byte packet every 10 second), shown in Figure 4. We see that all graphs look quite similar, and around 1–2 hours into the experiment, the smallest OWD drops to a new, stable level. The peak values are typical for mobile networks and originate from the ARQ mechanisms of the RA.

Now, we turn our attention to the minimum OWD obtained when varying the packet sizes, which is illustrated in Figure 5. There are two clear regions, one from 60 to 252 bytes and the other from 252 bytes and above. From the E220 GUI (MobilePartner), we get an indication of what service the operator is providing the modem with. However, the GUI only reports WCDMA or HSDPA as the service. When the experiments were conducted, all operators started by giving a WCDMA service. As the packet sizes increased they started to alternate between WCDMA and HSDPA, and eventually changed permanently to a HSDPA service. The point where the change was made varied a little bit

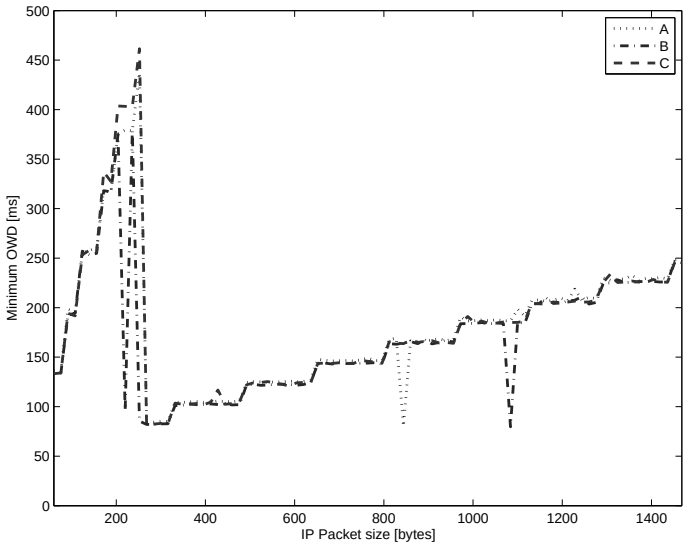

Fig. 5. Minimum OWD accross different operators.

from operator to operator, but it happened around 236 to 256 bytes, matching nicely with the drop of the OWD in Figure 4. Looking at the second region, we clearly see a staircase pattern. The steps are 144 bytes wide, and the step height is approximately 18 ms for all operators. This indicates that somewhere in the end-to-end path, some entity sends the data in blocks of approximately 144 bytes. In case of operator A and B, different behaviours were seen for two packet sizes, 844 bytes for operator A and 1084 bytes for operator B. The reason for this behaviour merits further investigation. It is also worth noting that both operators exhibited a very small minimum OWD around 80 ms, which is the smallest value in the HSDPA region.

In order to gain more insight into the behaviour of the minimal OWD at the transition between WCDMA and HSDPA regions, we conducted a second and more detailed experiment. The packet size was incremented from 100 bytes to 300 bytes with an increment of 4 bytes. The result is shown in Figure 6. The upper graph holds operator A and B, while the lower holds operator C. We have separated the graphs to highlight the patterns for operator A and B. For those operators, the staircase pattern is present, however not as pronounced as before. The width of the step is around 36 bytes, and the step height is approximately 60 ms. During the evaluation of the packets, both operators started by providing WCDMA service, then operator A changed (temporarily) to HSDPA for the 116 and 132 byte packets, while operator B didn't offer HSDPA until we reached a packet size of 196 bytes. But subsequently, both operators tend to offer HSDPA more frequently, and after 248 bytes both only offered HSDPA service. Looking at the measurements taken from operator C, that pattern is much less clear, the

operator had started switching to HSDPA earlier. Due to this, the same staircase pattern for packet sizes up to 188 bytes is not as clearly visible as for operators A and B.

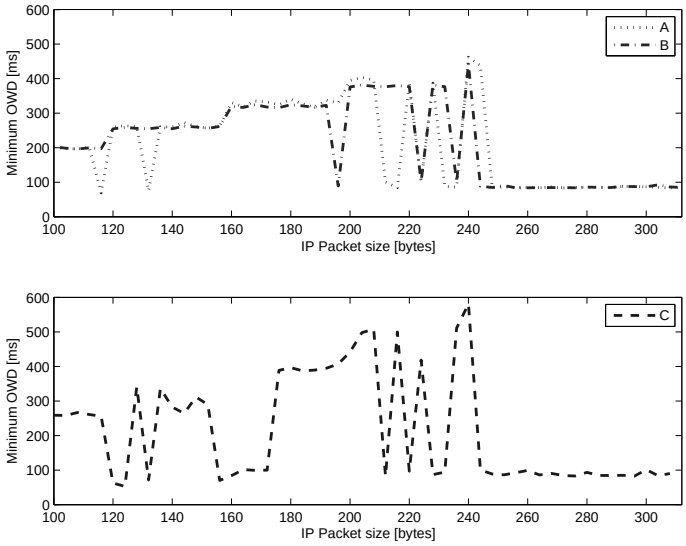

Fig. 6. Minimal OWD for packet sizes from 100 to 300 bytes.

5 Conclusions and Outlook

Based on quality-assured measurements from three Swedish mobile operators, this paper provides insights into the relationship between packet sizes and one-way delays, revealing the corresponding operator's resource allocation policy on the WCDMA/HSDPA uplink. Hereby, the quality of the one-way delay (OWD) measurements has been assured by a specific set-up and cabling scheme between Endace DAG cards that avoids common clock synchronisation problems, and by a quantification of the delay contribution of the gateway feeding the mobile link.

The most surprising result is that short packets might need more time to reach the receiver than long packets. Short packets experience a rather steep increase of OWD as their size is growing. For packet sizes in the range of 100 to 250 bytes, the minimal OWD varies heavily. For larger packets, it grows starting from quite small values (less than 100 ms) with a quite decent gradient. In both cases, the minimal OWD is increased approximately stepwise as a function of the packet size, which means that the maximal packet size per step allows for maximizing the throughput without paying for it in terms of extra delay.

Thus, our method and results deliver guidelines for application programmers to make the best out of mobile connectivity w.r.t. delay and throughput by choosing optimal packet sizes. In our case, packet sizes of at least 250 bytes avoid the potentially large and strongly varying miminal OWDs associated with smaller packets. Once again, the study shows the necessity to investigate the characteristics of network connectivities if these are not explicitly known. The outliers detected for operator A and B also merit further investigation.

Of course, it has to be observed that due to radio and network conditions, the actual OWD is likely to exceed the minimal value under consideration. Nevertheless, the minimal OWD indicates the best performance that can be expected, given the chosen packet size. The examination of further OWD statistics is left for future work.

The results obtained so far motivate the use of the proposed measurement method on the downlink and comparison of the results with each other and with measured roundtrip times. The quite significant delay for small packets might affect the effective throughput of downloads using TCP, as the acknowledgements are small packets carried on the uplink. Thus, further work will include the study of the impact of the discovered allocation policy onto TCP performance.

Acknowledgements

We would like to thank Ravichandra Kommalapati for conducting the experiments.

References

1. L. De Vito, and S. Rapuano, and L. Tomaciello: One-Way Delay Measurement: State of the Art, IEEE Transactions on Instrumentation and Measurement Vol: 57, No:12, Dec, 2008, pp 2742–2750.
2. D. Mills: RFC1305 Network Time Protocol (Version 3): Specification, Implementation and Analysis.
3. Endace Measurement Systems: http://www.endace.com, verified in January 2010.
4. S. Donnelly: High Precision Timeing in Passive Measurements of Data Networks, Ph.D. Thesis, The University of Waikato, 2002.
5. D. Veitch, S. Babu and A. Pásztor: Robust Synchronization of Software Clocks Across the Internet, Proc. Internet Measurement Conference, 2004.
6. P. Arlos: On the Quality of Computer Network Measurements, Ph.D. Thesis, Blekinge Institute of Technology, 2005.
7. Draka: SuperCat OUTDOOR CAT 5e U/UTP, http://communications.draka.com, verified January 2010.
8. James Messer: Ethernet FAQ, http://www.networkuptime.com/faqs/ethernet, verified January 2010.
9. P. Arlos, M. Fiedler and A. Nilsson: A Distributed Passive Measurement Infrastructure, Proc. Passive and Active Measurement Workshop, 2005.
10. D. Constantinescu, P. Carlsson, and A. Popescu: One-way Transit Time Measurements, Research Report, 2004, Karlskrona, Sweden.
11. High Level Design Description for Sunet: http://basun.sunet.se/aktuellt/OptosunetDesignv10.pdf, verified 2009-10-09.