Discovering parameter setting in 3G networks via active measurements

Antonio Barbuzzi, Fabio Ricciato, and Gennaro Boggia

Abstract—The behavior and performance of a UMTS network are governed by a number of parameter settings that are configured by the network operator, e.g., timeouts. In this letter we show that the actual value of such parameters can be inferred by a conceptually simple set of end-to-end measurements, without any cooperation with the network operator. In principle, such information can be used by researchers to define realistic network scenarios, e.g., for their simulations. Moreover, it can be used by a malicious attacker to fine-tune a large scale attack against the Radio Access Network, e.g., a paging attack.

Index Terms—Active Measurements, Cellular Networks, DoS

I. INTRODUCTION AND MOTIVATIONS

Third-generation (3G) mobile networks combine two different technology paradigms: the legacy cellular network model (e.g., GSM) on one hand, and the TCP/IP data protocols on the other. From the former they inherit a large degree of functional complexity, due to the need of guaranteeing efficient use of the logical and physical resource (Resource Management, RM) as well as seamless connectivity in mobility (Mobility Management, MM). An example of RM procedure is the assignment/release of a Dedicated CHannel (DCH) on the radio interface [1, p. 330], while an example of MM procedure is the paging [1, p. 406]. In general, RM/MM procedures require signaling interactions between the network and the Mobile Station (MS) and consume resources on both sides (bandwidth, memory, CPU, power). The algorithms triggering such procedures are typically very simple and often involve a single parameter, e.g., a timeout or a threshold, whose value is tunable by the network operator. In this work, we address the problem of inferring this value via end-to-end measurements.

We play the role of a network user, Bob, wishing to discover the value of some key network parameters without cooperation with the network operator. We see at least four possible motivations for Bob to carry out such investigation. He could be a researcher willing to set up a realistic 3G simulation scenario, with real parameter setting. Alternatively, Bob might be a staff member of a 3G mobile operator willing to discover the setting of other competitor networks, or instead checking that the actual parameter setting and equipment behavior of his own network are compliant with the planned values. Finally, Bob might be an attacker planning to launch a Denial-of-Service (DoS) attack toward the 3G network.

The latter scenario is probably the most intriguing and deserves some clarification. As noted already in [2] and [3], the functional complexity of the GSM paradigm, coupled with the “openness” of TCP/IP, exposes 3G networks to new security problems that only recently the research community has started to unveil. The most prominent example is the so-called “paging attack” (see [4], [3]): an attacker on the Internet contacts a large number of target MSs so as to trigger a high load of paging traffic in the radio network. If the attack is properly tuned, it can overload the capacity of the paging channels in the cells, thus impairing legitimate new connections. Similarly, it is possible to conceive a “DCH starvation” attack, where the attacker contacts many target MS at regular intervals slightly shorter than the DCH release timer, so as to prevent the release of the DCH and induce starvation of the available logical resources. Another form of attack considered in [5] is based on sending interval slightly longer than the DCH release timer. Following similar approaches, we believe that it is possible to conceive further forms of attacks, exploiting the deterministic nature of the mechanisms governing the RM/MM procedures.

In order to tune the attack pattern, the attacker would need to discover the exact value of some key parameters, e.g., the timers governing the channel transitions. In this letter we show that carefully designed end-to-end measurements can be used to discover the actual value of these parameters and at the same time to collect additional information about the network behavior. We present the principles of the measurement methodology and a set of preliminary results from two distinct operational networks located in different EU countries.

II. METHODOLOGY AND PRELIMINARY RESULTS

The measurement setting is depicted in Fig. 1. A wired host $W$ attached to the public Internet sends packets to a mobile host $M$ connected to the network under test with a 3G datacard. In our experiments, we used an Option GT FUSION+ HSDPA and an Option GT MAX 3.6 with Nozomi 2.1-2ubuntu1 driver. Both hosts were run on the same physical machine - an off-the-shelf Linux PC - so that a single clock is used. All packets departing/arriving on both interfaces (wired and wireless) are captured and timestamped with tcpdump, and one-way delays are computed as the difference in the arrival times. The accuracy of the PC clock is in the order of 0.1-0.2 ms, which is enough for our purposes as the typical delays through a 3G network is in the order of tens of milliseconds. Note also that the delay component in the wired section is typically much smaller than the delay in the 3G section and, most importantly, it is largely...
EU countries, hereafter referred to as Net.1 and Net.2. the observed dependency of the total end-to-end delay with the sending pattern. Most of our experiments involve downlink packets with IDT slightly below the value of \(d\). Obviously it holds that \(B_{DCH} \gg B_{CCH}\). At point \(d\), the buffer is again empty. After this point the horizontal distance between the two curves represents the downlink delay in the DCH channel, denoted by \(d_{DCH}\).

In summary, the geometric analysis of the graph in Fig. 2 enables the estimation of several system parameters. A certain estimation error is present due the delay jitter introduced by the wired network, but the accuracy can be improved by averaging over multiple experiments. To avoid lengthy manual analysis of the graphs, we used simple heuristics to automatically identifies the “knees” of the piece-wise process \(S_M(t)\) and extract the length and location of the various segments.

The DCH release is based on a timeout that is reset upon each packet transmission (sent or received) by the MS. The value of the timeout, denoted hereafter by \(T_{DCH,off}\), is tunable by the operator. The experiment to discover its value is conceptually simple: we first force the MS to acquire a DCH by sending an initial high-rate burst. This is basically the same pattern discussed above in Fig. 2 until point \(d\). Then we send a sequence of packets with slowly increasing inter-departure time (IDT). Formally, the \(k\)th IDT is given by \(\tau_k = \tau_{k-1} + \Delta\tau + \omega\), where \(\Delta\tau\) is a fixed increment and \(\omega\) a small randomization term uniformly distributed in \([-\Delta\tau/2, +\Delta\tau/2]\). We refer to such pattern as “chirp”, in analogy to some radar techniques. The randomization prevents measurement bias, and turns useful when merging data points of different experiments in the same plot to avoid superposition of data points. At some point during the chirp, the MS switches back to CCH. The last value of the IDT is then taken as an estimate for \(T_{DCH,off}\). The CCH transition can be recognized in two ways. First, some 3G cards/drivers report explicitly the channel status. Alternatively, one can look at the one-way delay packets of the secondary sequence for different values of the IDT: as \(d_{DCH} \ll d_{CCH}\), the sharp transition point marks the DCH-to-CCH switch. In Fig. 3 we report the resulting graphs for both tested networks (vertical log-scale). They reveal a value of \(T_{DCH,off}\) around 5 and 2.7 seconds for Net.1 and Net.2, respectively. We observe that the packets arriving exactly during the channel switch experience very large delays, around 2 seconds. This can be taken as an estimate of the completion time of the channel switch procedure. We verified experimentally that sending a continuous low-rate stream of packets with IDT slightly below the value of \(T_{DCH,off}\) forces
the DCH to remain assigned to the MS indefinitely. This suggests the possibility of launching DCH starvation attacks.

In the next set of experiments we aim at discovering the value of the timer governing the paging. If the MS is silent for more than $T_{PCH}$ seconds, it is instructed by the RNC to switch to the Paging Channel (PCH). If then a downlink packet arrives for this MS, the network must start a paging procedure in the whole Routing Area to learn its current cell and to transfer the packet [1, p. 332]. We denote by $D_{pag}$ the time required to complete a paging procedure (paging completion time). The end-to-end delay of the front packet (triggering the paging) includes the paging delay. If the first paging fails the network repeats a new paging requests after $T_{rep}$ seconds, which further increases the total downlink delay of the front packet. In order to discover the value of the $T_{PCH}$, we set the MS on CCH and send a “chirp” pattern with initial IDT slightly larger than $T_{DCH,off}$. The initial chirp packets experience a delay equal to $d_{CCH}$ as long as the inter-departure time stays below the (unknown) value of $T_{PCH}$ (refer to Fig. 4). After this point, the probe packets will trigger paging and their end-to-end delays increase considerably due to the paging delay component. The change-point marks the actual value of $T_{PCH}$. The delay gap between the two data point clusters represents an estimate of the paging completion time, denoted by $D_{pag}$ (see Fig. 4).

In Net.2, we observe that for IDT in the range 28.6–36.8 seconds the probe packets experience alternatively high and low delays. To explain the phenomenon we ran a number of additional experiments - not shown here for space constraints - that led to two interesting discoveries. First, we found that in Net.2 every paging procedure is followed immediately by a DCH assignment regardless of the actual following traffic. If no packet follows the paging, the DCH is released after $T_{DCH,off}$ (measured earlier). The second finding is that the timer $T_{PCH}$ is restarted upon DCH release. The combination of these events produces the dynamic shown in Fig. 5, where it can be seen that for IDT in the range $[T_{PCH}, T_{PCH} + D_{pag} + D_{DCH, on} + T_{DCH, off}]$ only one packet out of two triggers paging. This is one example out of a number of unexpected findings that we encountered during our measurement campaign, often pointing to particular dynamics internal to the network equipments.

![Fig. 5. Time diagram of paging events.](image)

### III. DISCUSSION AND FUTURE WORKS

In Table I we summarize the estimated values of the most relevant parameters for both tested network (ranges indicate 95% confidence intervals). A direct comparison indicates that Net.2 has globally better delay performances than Net.1 - at least in the tested location. One could repeat the experiments at several different locations (cells) to gather a more comprehensive view of the network-wide performance.

During our experiments, we had to cope with the fastidious presence of so called “unwanted traffic” [6], i.e., unsolicited packets arriving from the Internet. Such packets are received in between probes packets, thus breaking the regularity of the IDT sequence within the chirp, leading to an invalid delay sample. The problem is particularly critical for experiments with large IDT, e.g., for $T_{PCH}$ estimation. The frequency of unwanted packets was very high in Net.1 and almost negligible in Net.2. In fact, the latter is configured to block a few ports known to be used mainly by malware (e.g., tcp:445,135), while it seems that Net.1 does not filter any port. This again suggests a somehow less mature setting of Net.1 compared to Net.2.

The possibility of performing such measurements is bound to the deterministic nature of the parameter setting. As part of our future work on countermeasurers, we intend to explore the role of RM/MM randomization (e.g., timers, thresholds) to increase the network robustness to deliberate attacks.

### REFERENCES