Traffic Analysis at Short Time-Scales: An Empirical Case Study From a 3G Cellular Network

Fabio Ricciato, Eduard Hasenleithner, and Peter Romirer-Maierhofer

***Abstract*—The availability of synchronized packet-level traces captured at different links allows the extraction of one-way delays for the network section in between. Delay statistics can be used as quality indicators to validate the health of the network and to detect global performance drifts and/or localized problems. Since packet delays depend not only on the network status but also on the arriving traffic rate, the delay analysis must be coupled with the analysis of the traffic patterns at short time scales.**

In this work we report on the traffic and delay patterns observed at short timescales in a 3G cellular mobile network. We show that the aggregate traffic rate exhibits large impulses and investigate on their causes. Specifically, we find that high-rate sequential scanners represent a common source of traffic impulses, and identify the potential consequences of such traffic onto the underlying network.

This case-study demonstrates that the microscopic analysis of delay and traffic patterns at short time-scales can contribute effectively to the task of troubleshooting IP networks. This is particularly important in the context of 3G cellular networks given their complexity and relatively recent deployment.

***Index Terms*—Network monitoring, traffic analysis, network management, cellular networks.**

I. Introduction

THE past five years have witnessed the deployment of third-generation (3G) mobile cellular networks in several countries. Such systems have extended the reach of Internet access, allowing anywhere-anytime wireless access to a large population of nomadic users equipped with portable terminals, mainly laptops and smart handheld devices. Besides the specific aspects related to mobility and radio resource management, a 3G cellular network is ultimately a large IP access network, similar in many ways to large enterprise networks. These systems are generally complex to manage and maintain: they typically include many different kinds of equipments and technologies and are subject to continuous updates and upgrades. Furthemore they are exposed to an ever-changing "usage environment" as preferred applications, services and mobile terminal capabilities evolve. Operating and troubleshooting such networks is a challenging task. The key challenge is often to promptly detect anomalies, hidden performance degradations and potential problems before they lead to major outbreaks. A powerful approach to achieve such goals relies on continuous traffic monitoring and analysis of packet-level traces. Passive monitoring does not require direct access to the network nodes and is therefore completely equipment-independent: this aspect represents an enormous operational advantage if compared to the traditional approach to network monitoring that relies exclusively on data provided by the network elements themselves (logs, counters). Extensive traffic monitoring can be operated cost-effectively nowadays also at multi-Gbps speed, given the availability of specialized acquisition hardware (e.g. wiretaps, DAG cards [1]) and large storage solutions at accessible cost.

Traffic monitoring and analysis is now an important topic of research. The ability to identify what to measure and how to interpret the results, i.e., to "read the traffic", particularly within the perspective of network validation and troubleshooting, must rely on extensive and deep knowledge of the traffic environment in the real network. Such a knowledge basis is now being built collectively by the research community. In this study we provide a contribution in this direction by reporting a set of observations from a real network.

The initial motivation for this work was to extract one-way delay measurements from synchronized traces captured at different network sections, and to use the delay statistics to detect performance drifts or hidden problems between the measurement points. The emergence of delay values higher than the "physiological" level observed in the past can be taken as the symptom of capacity shortage or even misfunctioning of some intermediate network element. Both cases would require attention by the network operation staff. We are adopting this approach in practice, by implementing additional delay processing features to a monitoring tool that was developed in a past project [2]. The system is now deployed in the operational GPRS/UMTS network of a major mobile operator in Austria. The ultimate goal is to build a module that can be used for daily network operation and troubleshooting. As a preliminary study, we have analysed off-line the delay and traffic patterns at several network interfaces during one full day. In this paper we report on our experience during this preliminary study, discussing some methodological aspects and presenting our findings in detail. Specifically, we focus on the empirical analysis of the traffic rate and delay process at short time-scales, where we observe impulsive patterns, and investigate their causes. Most of these findings are not specific to mobile networks but rather applicable to any IP-based access network. We believe that this work offers a valuable contribution to the community of experts towards a more complete understanding of the traffic process in the modern Internet. The case-study presented here demonstrates that the microscopic analysis of delay and traffic patterns at

Manuscript received October 30, 2007; approved by IEEE Transactions on Network and Service Management Editor Marcus Brunner.

E. Hasenleithner and P. Romirer-Maierhofer are with the Forschungszentrum Telekommunikation Wien (FTW) Donau City Straße 1, A-1220 Vienna, Austria, EU.

F. Ricciato is with the University of Salento, Lecce, Italy and with FTW (e-mail: {ricciato, romirer}@ftw.at).

Digital Object Identifier 10.1109/TNSM.2008.080102.

short time-scales can be a valuable tool for troubleshooting in real networks.

The rest of the paper is organized as follows. In §II we provide a short description of the network under study and of the monitoring tool. In §III we present a set of measurement results from the real network, specifically on the delay and traffic rate observed at small time-scales (subsecond). A major finding is that both processes contain impulses (spikes) caused by high-rate scanning sources. In §IV and §V we elaborate on the impact of the scanning traffic in the Core Network and in the Radio Access Network, respectively. The content of both these sections is based on the analysis of traces from the operational network. In §VI we relate our study with previous literature. In §VII we discuss our findings and the lessons to be learned from the present work. Finally in §VIII we conclude.

II. Monitoring setting

A. *Structure of a 3G mobile network*

The reference network structure is sketched in Fig. 1. The Mobile Stations (MS) are connected via radio link to the antennas. In our network four different access schemes are possible depending on the geographical location of the MS and its terminal capabilities: GPRS, EDGE, UMTS and HSDPA [3]. A set of geographically neighboring antennas is connected to a *controller*, called Base Station Controller (BSC) in GPRS/EDGE and Radio Network Controller (RNC) in UMTS/HSDPA. These are then connected to a set of so-called Serving GPRS Support Nodes (SGSN) via the Gb (for GPRS BSC) and IuPS (for UMTS RNC) interface. The overall set of antennas, BSC/RNC and the links to the SGSNs constitute the Radio Access Network (RAN). The primary role of the SGSN is to perform mobility management, which involves frequent signaling exchanges with the MSs. In a typical network, there are several SGSNs located at different physical sites. The data-plane traffic collected by the SGSN is concentrated within a small set of so-called Gateway GPRS Support Nodes (GGSN). The GGSN acts as the IP-layer gateway for the user traffic: it is in charge of setting up, maintaining, and tearing down a logical connection to each active MS, called "PDP-context", that is conceptually similar to a dial-up connection. During the set up of the PDP-context, an IP address is dynamically assigned to the MS. The set of SGSNs and GGSNs is interconnected by a wide-area IP network that will be referred to as the "Gn network" (ref. Fig. 1) following the terminology of 3GPP specifications ("Gn interface"). In the Gn network, the IP packets coming from / directed to each MS are tunnelled into a 3GPP specific protocol (GPRS Tunnelling Protocol, GTP [3]) and then encapsulated into an IP packet travelling between the SGSN and GGSN. After the GGSN, the user packets enter into a network section that is functionally similar to the Point-of-Presence (POP) network of an Internet Service Provider (ISP). We will refer to this network as the "Gi network". Similarly to any enterprise network, the Gi section is delimited by one or more Edge Routers (ER in Fig. 1) and includes a number of IP-based elements: servers, WAP gateway, proxies, DNS, firewalls, etc. In the network under study, part of the WEB traffic is handled by a transparent proxy. Also, *public* IP addresses are assigned to the MSs, i.e., Network Address Translation is not used.

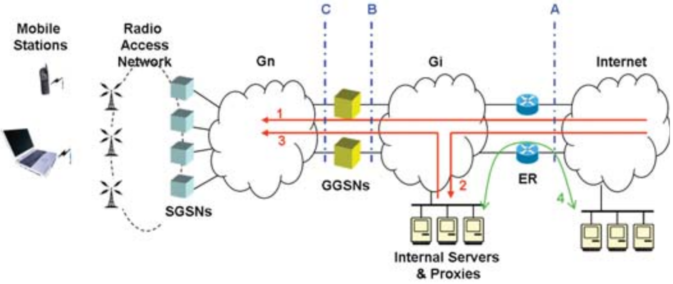


Fig. 1: Network structure and monitoring setting.

B. *The monitoring system*

The present work is based on packet-level traces captured in the operational network of a major mobile provider in Austria, EU. We monitored the links at three different interfaces (ref. Fig. 1): the external peering links (point "A") connecting the Edge Routers to a national ISP, the Gi ("B") and Gn ("C") links near the GGSNs. All the GGSNs and the ERs co-located at a single physical site were monitored, corresponding to a fraction x of the total network traffic (undisclosed [1]).

The monitoring system was developed in a past project [2]. The capture cards are Endace DAG [1] with GPS synchronization. For privacy reasons, we store only the packet headers up to the transport layer, i.e., payload is stripped away. All packets are captured and no sampling is implemented. On the Gn interface, the system is capable of parsing the GTP layer and tracking the establishment and release of each "PDP-context", and to uniquely identify the MS sending or receiving each packet. Similarly to timestamps, a unique MS identifier is stored as an additional label information for each frame. To protect the user privacy, the MS identifiers are chosen as arbitrary strings, decoupled from the real MS identity.

For this work, we analysed the complete traces from all the three interfaces (peering links, Gn, Gi) during one full day in September 2006. For each interface we monitored multiple links attached to different elements. For the sake of simplicity we consider only the downlink traffic in this work (directed to the MSs). Additionally, we monitored the Gb and IuPS links of two distinct SGSNs, where we collected two half-day traces including both signaling and data packets. From these we extracted the rate of paging requests presented later in §V.

C. *Delay computation*

The one-way delay was extracted in post-processing with a similar methodology as in [4]. For each packet, some selected fields of the IP header (port and IP addresses, identification field) plus the whole TCP header are hashed into a string of length $N = 128$ via the MD5 function [5] (instead CRC-32 was used in [4]). This guarantees a negligible collision probability. If the first K bits of the hashed value match a predetermined K-mask (e.g. all zeros) the packet is picked as a "candidate sample" for delay calculation. This corresponds

[1] Several quantitative values are considered business-sensitive by the operator and subject to non-disclosure, e.g. absolute traffic volumes, number of active MS, number and capacity of monitoring links. For this reason the following graphs reporting absolute traffic values have been rescaled by an arbitrary undisclosed factor.

to a sampling rate grossly equal to $1/2^K$ given the good uniformity behavior of the hash function. For each candidate sample, the hashed string, the arrival timestamp, and both IP addresses are output to a separate binary file. The same procedure is repeated for the traces captured at each interface, using the same K-mask. In order to extract the delay samples from A to B, we seek for string matches in the two different traces, and take the difference of the respective timestamps. While this scheme is conceptually simple, there are a number of practical issues that must be taken into account.

Missing matching. First, a hashed value observed at B (Gi links) might have no match at A (peering links). This might indicate that the packet was lost on its way from A to B, or that the packet arrives at B from a path that does not include interface A. The latter case includes all traffic generated by a source internal to the Gi network, e.g. a DNS reply, or a WAP connection. The part of WEB traffic that is handled by the transparent proxy does not produce delay samples because the associated IP flows are "divided" by the proxy that modifies the TCP/IP headers (flows 2 and 3 in Fig. 1), hence the related hash value. When no match is found, no delay sample is produced. This means that the actual rate of valid delay samples r will be lower than the theoretical rate, i.e., $r = \frac{\alpha}{2^K}$ with $\alpha < 1$. Note that WEB and WAP constitute the dominant components of 3G traffic (see [6]). In our traces, we measured $\alpha \approx 50\%$.

Multiple matches. In some cases, a single hashed value observed at A has multiple matches at B. We recognized that this is due to *routing loops*. Consider what happens when a packet traversing A towards B enters into a routing loop between the GGSN and any intermediate router within the Gi network. Assuming a loop of length 2, the looping packet will be observed once at A (say with $\texttt{TTL}{=}M$) and $(M-1)/2$ times at B, each time with a different $\texttt{TTL}$ value $(M-1, M-3, M-5, ...)$. However, only the first observation holds a meaningful delay sample. It is not difficult to identify the presence of looping packets in the traces and to filter them out for the delay extraction. Incidentally, we note that a simple processing module specialized on revealing routing loops can be a small but usefulsupport for network operation.

III. Observed patterns

A. Delay measurements

We start by presenting the one-way delay measured between the peering links and the Gi interface (points A and B in Fig. 1). In Fig. 2, we plot the delay samples measured within two different intervals of 1 hour, representative of low and moderate load, along with the Empirical Complementary Cumulative Distribution Function (ECCDF) for each interval. As expected most of the samples take very low values, below 1 ms. However, we also observe large delay values, up to 200 ms. The high delays in Fig. 2a and 2b are not persistent but appear concentrated into vertical lines that are scattered uniformly across the whole day. The simplest hypothesis is that they are associated with traffic bursts. Similarly, in Fig. 3, we report the delay measurement between points B and C (ref. Fig. 1). The network section in between includes only the GGSNs. Similarly to A-B, B-C delays also stay mostly below 1 ms and show only sporadically high delays, again concentrated into vertical clusters (*delay spike*). Note that the density of large delay samples in the B-C section is lower than in A-B. Moreover, the delay spikes in B-C always have a corresponding delay spike in A-B (the inverse is not true). The latter observation confirms the hypothesis that large delays are due to external traffic burts rather than internal equipment dynamics.

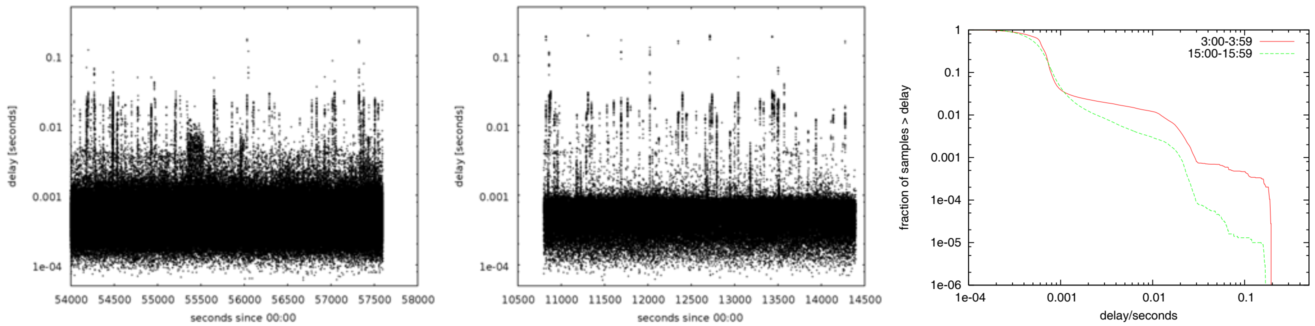
B. Rate measurements

In Fig. 4, we report the downlink traffic rate arriving at each interface. The rate is measured as byte counts in timebins of 1 second (the graphs are arbitrarily rescaled, see note [1]). All figures yield the typical time-of-day behaviour found in any public network: the traffic mean and variance are lower at night, where less users are active, and reach their peak in the late evening. In addition, we noticed the presence of *traffic spikes* (positive impulses) scattered across the whole day on the peering links and on Gi. We manually inspected the causes of the traffic spikes arriving at each interface, and in the next sections we provide a detailed description of each cause. In Figures 4b and 4c, we also note the presence of *traffic notches* (negative impulses) on Gi and Gn, an aspect that will be discussed below.

C. High-rate sequential scanners

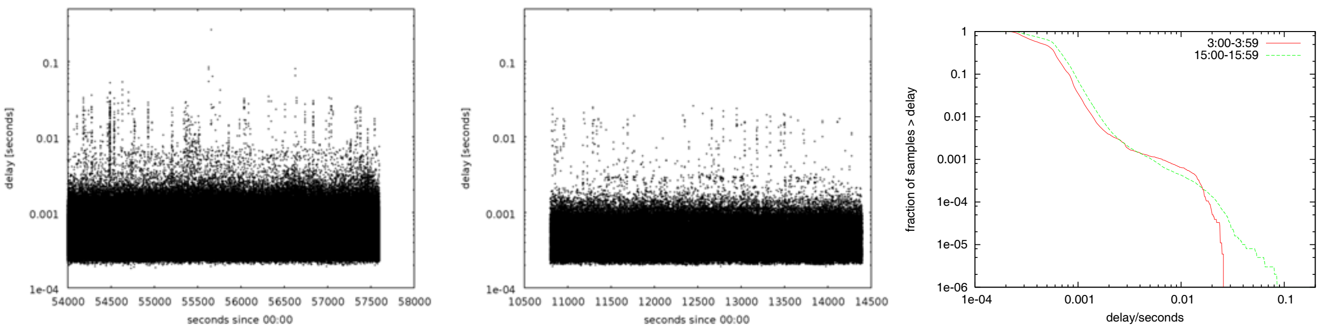
We found that the primary cause of traffic spikes at the peering links are *scanners*. Consider a host in the Internet that is performing a *sequential scanning* of the full address space by sending some kind of probe packet (e.g. ICMP, TCP SYN, UDP) at the rate of R probes/sec. At some point it will start spanning the address block assigned to the local network, say of size L (e.g. $L = 2^{12}$ for a /12 block). This will result on an impulse of incoming traffic lasting L/R seconds. If we count the traffic in time bins of lentgh $\Delta > L/R$ and assume that the whole burst falls within a single time bin, the packet count in this bin will jump by L. If the address space allocated to the network consists of several non-contiguous blocks, we will observe a different spike for each block. The height of each spike equals the width of the block, and the separation in time directly relates to the distance in the address space between the blocks. In other words, the probe traffic generated by a sequential high-rate scanning source causes an arrival rate pattern at the peering link that mirrors the address space allocation to the local network. If the scanner source keeps cycling into the address space, such patterns will appear periodically.

We observed exactly this phenomenon in the real network: in Fig. 5a we report the arriving packet rate measured at the peering links for the traffic sent by a single external IP address identified as a high-rate sequential scanner. In the figure, packets are counted in timebins of 1 minute. In practice, the probe packets arrive clustered into bursts of very short duration, corresponding to a relatively high bitrate (non disclosed). We observed several high-rate scanners, periodic and not, using different packet types (e.g. TCP SYN, UDP) to different ports. If the scanning occurs on a blocked port (firewalled), the probe packets will be immediately discarded



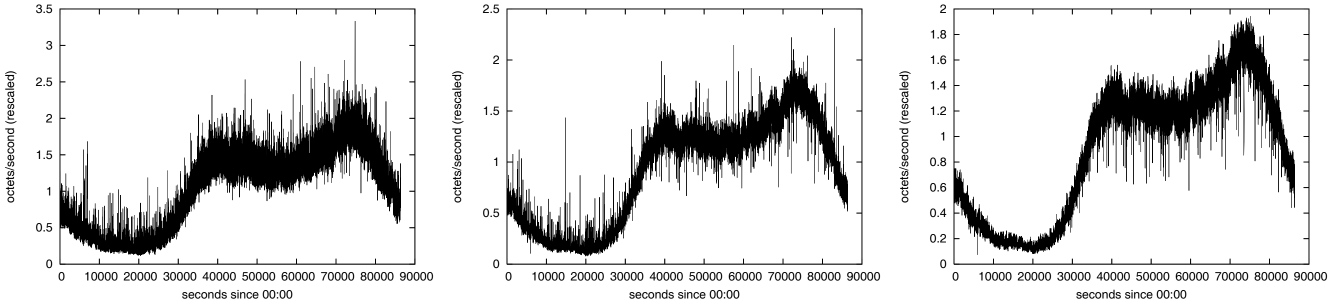
(a) Day time (3-4 pm) (b) Night time (3-4 am) (c) ECCDF

Fig. 2: Downlink delays in the Gi network, between the peering links and Gi links (A-B section).



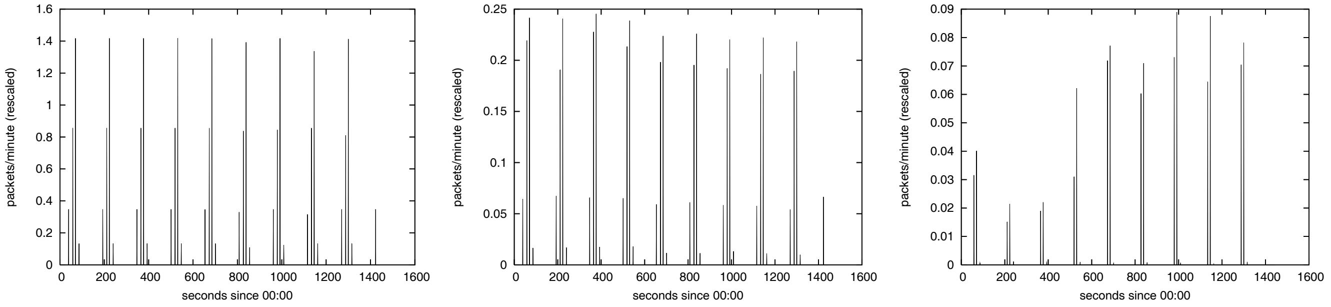
(a) Day time (3-4 pm) (b) Night time (3-4 am) (c) ECCDF

Fig. 3: Downlink delays internal to the GGSNs, between Gi and Gn links (B-C section).



(a) Peering links (b) Gi links (c) Gn links

Fig. 4: Total downlink packet rate (rescaled), timebins of 1 second, one full day.



(a) Peering links (b) Gi links (c) Gn links

Fig. 5: Downlink packet count (rescaled) for a single high-rate sequential scanner (S1 source), timebins of 1 minute, one full day.

at the Edge Router. Therefore, no associated traffic spike will be observed at Gi. On the other hand, in case of non-blocked ports, the scanning burst will penetrate into the Gi network. At the time of measurement, we verified that most of the incoming high-intensity scanning occurred on a single UDP port (undisclosed). Note that some other ports that are commonly used by scanning worms and various malware were already filtered by the firewall.

In Fig. 5b, we report the packet rate from S1 as observed at Gi. Comparing Fig. 5a and 5b (note the different vertical scales), we find that the spikes are lower at Gi than on the peering links: at the peak, only approximately 20% of the probing packets reach the GGSN, the rest being lost somewhere along the A-B path. The microscopic analysis of the delay patterns revealed that the loss of probe packets is caused by micro-congestion in the A-B path caused by the the scanning traffic itself. Comparing 5b and 5c, we observe that in each time bin only a fraction of the packets observed at Gi passes the GGSN. In fact, the GGSN forwards only the packets directed towards active IP addresses, i.e., currently assigned to a MS within an active PDP-context. The number of packets passing the GGSN provides a rough estimate of the number of contemporary active PDP-contexts within each block (in order to to hide this information, we had to rescale the graphs in Fig. 5 by an undisclosed factor). This is lower at night when most MSs are inactive. Therefore, the penetration of scanning traffic is subject to a time-of-day effect, and is maximum at the peak-hour. In the specific case shown in Fig. 5, we found that a small fraction of MS responded to each packet with a ICMP "port unreachable" message, thus causing backscatter traffic in the uplink direction. We remark that S1 was just one of several sequential scanning sources with patterns similar to Fig. 5, periodic and not, that were observed in the traces. Moreover, we also found that other sources were performing *(pseudo-)random scanning* for long periods (whole days). In this case, the visits to each address block are scattered across the whole scanning period instead of being concentrated into distinguishable spikes (as with sequential scanning). Hence the rate of arriving probes is steady, with small fluctations around a relatively low average value that remains constant during the whole scanning period. As a result, random scanning is much less invasive then sequential scanning from the perspective of an individual target network.

IV. Impact of scanning traffic in the Core Network

A. Micro-congestion

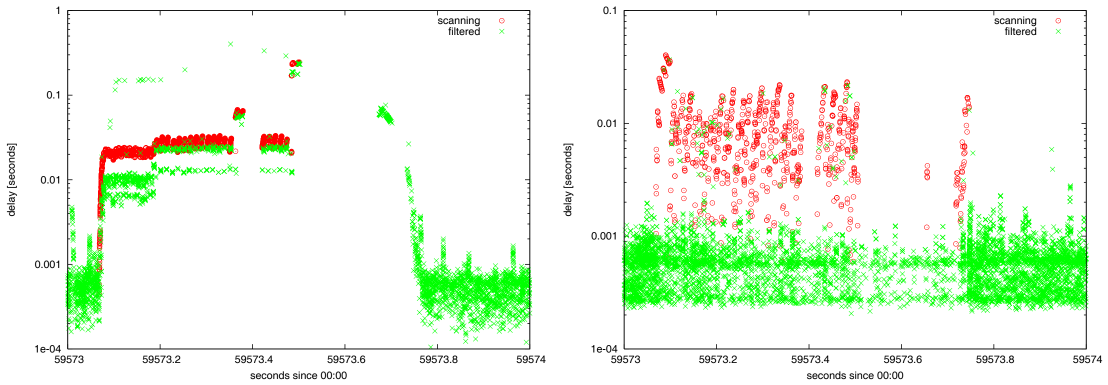
The simplest explanation for the loss of scanning packets in A-B is that the probe bursts were clipped after hitting the capacity limit of some internal resource, e.g., a link or the CPU of some router. To verify this hypothesis, we zoom into a sample scanning burst from S1 and extract a complete ($K = 0$) set of delay measurements in its neighborhood. We divide the A-B delay samples into two groups: "scanning" packets (identified by the IP address of source S1) and other traffic ("filtered" timeseries). The delay samples for both groups are shown in Fig. 6a. For each packet arriving at A at time t_A and observed at B at time $t_B > t_A$, we mark a point of coordinates $\langle t_A, t_B - t_A \rangle$. Let us focus on the "scanning" time-series: the delay pattern is consistent with the presence of a buffer that fills up rapidly (initial slope) and then remains persistently saturated (plateau). After the fill-up phase, most arriving packets are lost, while others gain access to the (almost) saturated buffer and experience an approximately constant delay equal to the buffer depletion time, about 20 ms in this case. Such a delay pattern is consistent with the hypothesis that some link is being saturated for a short period by the scanning burst along the A-B path (micro-congestion). The exact peak rate of the scanning packets arriving at Gi (undisclosed in our case) would provide an immediate indication about its bandwidth to the network staff. This can facilitate the localization of the (micro-)congested link.

After this phase, in Fig. 6a we observe a *cluster* of considerably larger delay samples, around 200 ms, followed by an empty period of approximately 200 ms where no packets are received at all. This pattern is consistent with the so-called "coffee-break" event [4], i.e., a temporary interruption of the packet forwarding process at some intermediate router. In our traces, such events only occurr during large scanning bursts, suggesting that the coffee-breaks observed in this network are *not* due to "normal" router dynamics as in [4] but rather a symptom of short-term CPU congestion. Note also that in Fig. 6a, the delay pattern just described for the "scanning" packets is followed by the remaining traffic as well ("filtered" series). This indicates that the micro-congested resources (buffer, CPU) are shared by all traffic. In other words, the scanning traffic is causing a small impairment to the other legitimate traffic, causing micro-congestion and temporarily high delay and loss. However such events last a very short time, typically a fraction of a second, and remain invisible to individual users, at least as long as they do not occurr too frequently.

In Fig. 6b, we report the delay samples for the B-C section, i.e., internally to the GGSN. Recall that only a fraction of the scanning packets pass through the GGSN, specifically only those hitting an IP address currently allocated to an active MS. Therefore, we have less delay samples for the scanning traffic. It can be seen that the scanning packets experience higher delays than the rest of the traffic. Differently from the A-B section (Fig. 6a), that most of the other traffic on B-C seems unimpaired by the presence of scanning, indicating that the two traffic components access different resources inside the GGSN. Notably, only a few non-scanning packets yield high delay during the scanning bursts: these are packets opening new connections (e.g. TCP SYN). The most likely explanation is that the GGSN embeds stateful mechanisms, with a separate internal data path for the connection-opening packets. Clearly, this is also accessed by scanning bursts. In this specific case, such a data path is only mildly affected by the scanning load (higher delays, no loss).

B. Uplink packing

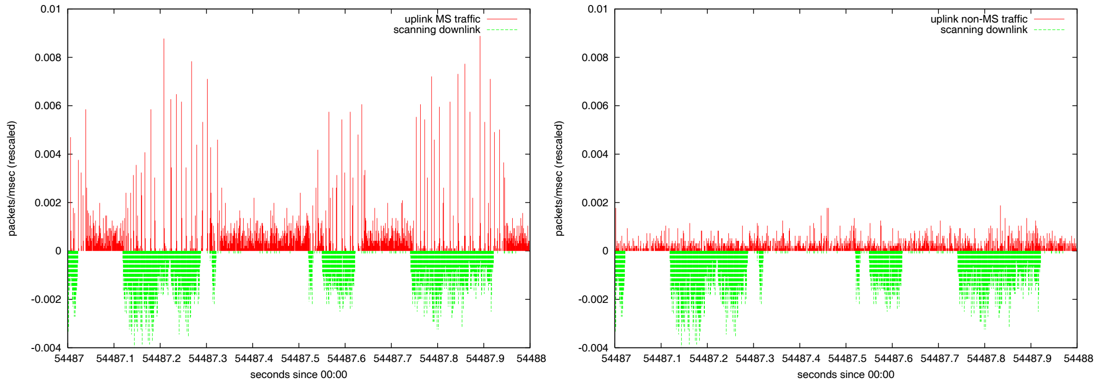
We noticed that the presence of scanning traffic in downlink had an impact on the traffic pattern in *uplink* as well. Let us now consider the uplink traffic (towards the Internet) observed at the peering links. This consists of two components: the traffic arriving directly from the MSs via the GGSN ("MS



(a) Delay between peering links and Gi (A-B section).

(b) Delay between Gi and Gn links (B-C section).

Fig. 6: Downlink delay pattern during a scanning burst: scanning probes (red 'o') and remaining packets (green 'x').



(a) MS traffic

(b) Server traffic

Fig. 7: Correlation between uplink traffic and downlink scanning: packet counts in 1 ms bins at peering links (rescaled).

traffic"); and the traffic originated by the internal servers ("server traffic"). The latter shares only a part of the A-B path with the MS traffic (ref. Fig. 1). On the peering links, we can distinguish the two components based on the source IP addresses, because the MSs are assigned specific address blocks in the network under study.

In Fig. 7a, we report the packet count for the "MS traffic" in time bins of 1 ms during one second. On the negative axis, we draw the count of scanning packets observed at the peering links in downlink. Similarly, Fig. 7b reports the uplink pattern for the "server traffic". From Fig. 7a, we see that the uplink "MS traffic" behaves very differently during the scanning compared with periods of scanning silence: the uplink packets are transmitted into discontinuous bursts, corresponding to large spikes in the packet count that are well separated from each other, with no transmissions in between. Instead, in the absence of scanning traffic, the uplink rate tends to vary more continuously. This indicates that the downlink scanning bursts at some internal router (call it "node X") has an impact on the forwarding process of the uplink packets as well. It is possible that the CPU scheduling at node X lets the scanning traffic starve the uplink forwarding process for a few milliseconds. During this period, the arriving uplink packets are buffered, and when the CPU returns to serve the uplink queue all these packets are forwarded at once, "packed" into a single burst. Note that most of the uplink traffic consists of ACK packets of TCP flows, resulting potentially in a sort of ACK synchronization at small time-scales. At this point, it is possible to speculate that, in principle, such an effect might in turn produce synchronization effects and fluctuations on the TCP aggregate, or at least on some of its component (e.g. traffic from a major server). Notably the relationship between ACK compression and TCP fluctuation was established in [7]. In the specific case under study, we did not find any evidence that the observed "ACK packing" translates into synchronization of TCP flows, probably because of the large Round-Trip-Times (RTT) values and heterogeneity, up to hundreds of milliseconds in this network (for more details see [8]).

Finally, comparing Fig. 7a and 7b, we observe that the uplink "server traffic" is not affected by the presence of scanning activity. This indicates that the node X is located on the section of A-B path that is *not* shared by server traffic, thus providing another hint for the localization of this element. It is remarkable that it is possible to collect such information without any detailed knowledge of the exact structure and deployment of the Gi network.

C. *Rate notches*

Recall from Fig. 4b and 4c that *notches* (negative impulses) are present at Gi and Gn in the downlink rate. Further inves-

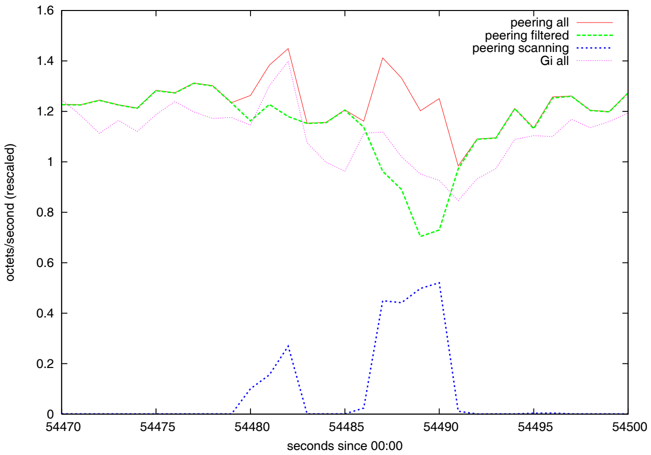


Fig. 8: Zoom into a traffic notch on Gi.

tigations into a few sample cases revealed a close correlation with the scanning process. To see this we draw in Fig. 8 the total downlink traffic rate arriving at the Gi links ("Gi all"). A notch is evident at the center of the figure where the instantaneous rate falls by approximately 40% from the local average. In the same graph, we report separately the two traffic components arriving at the peering links, namely "scanning" traffic from source S1 and all other traffic ("filtered"). It is now evident that the notch at Gi corresponds to a lower rate of legitimate traffic arriving at the peering links, which in turn is caused by a spike of scanning traffic. It is remarkable that the total traffic rate arriving at the peering links ("peering all") does not display an appreciable variation since the scanning spike and the notch compensate each other to some extent. The scanning traffic can locally reduce the arrival rate of legitimate traffic in two ways. First, by generating micro-congestion inside the network, it generates synchronized packet loss for many TCP flows, that the TCP closed-loop will translate into a reduction in the next traffic arrival. Second, it is possible that the scanning bursts also cause micro-congestion externally to our network, i.e., in the neighboring ISP, leading to loss of legitimate packets before the peering links.

D. *Evoking latent routing loops*

We found that some of the traffic spikes seen on Gi links (ref. Fig. 4b) were due to packets trains bouncing in a routing loop terminated at the GGSN. The loop was affecting only the static route to a small block within the internal address space. It appears that this block was not in use, i.e., not included in the pool dynamically assigned to the MSs. As a result, the routing loop remained "latent", with no traffic for most of the time. Curiously, this routing loop was regularly evoked only by the scanning traffic: any sequential scanner would at some point span this specific block, causing a full train of probe packets to enter the loop at the same time. The routing loop was amplifying the "shot" of traffic burst from the scanner. It is natural to ask what impact this routing loop had onto the other traffic. To answer this question, we can again resort to the A-B delay analysis in the neighborhood of a traffic spike associated with a train of looping packets. In this case, it appeared that the delay process of the other traffic remained unaffected by the looping burst. Notably, this observation suggests that the "node X" experiencing micro-congestion as discussed in §IV-A must be located before the other end-point of the loop, thus providing a further indication for its localization.

Routing loops caused by routing transitories in the backbone network are extensively investigated in [9]. They must be considered as "physiological" events in large networks with dynamic routing. This does not apply here, where the loop discovered was persistent and ultimately due to a small configuration error.

V. Impact of scanning traffic in the Radio Access Network

In a previous work [10] we showed that Internet background traffic can cause a waste of resources in the radio section of 3G cellular networks. The problem discussed there involves the timer-based mechanism for the dynamic assignement and release of Dedicated Channels (DCHs) on the radio interface. Under certain conditions, background traffic can prevent the DCH from being released. In the following, we present empirical evidence of another effect of background traffic, and specifically of scanning traffic, onto the Radio Access Network (RAN). The problem directly affects the paging traffic on the packet-switched domain.

We recall some background of cellular networks. Let us first consider the case of GPRS. The RAN is divided into cells which are organized into so-called Routing Areas (RAs). At a generic instant, each MS can be in one of three "mobility states" (or "modes"): IDLE, STANDBY and READY. The state transitions events are sketched in Fig. 9 (see [3, p.129-131]). When attached to the network, the MS can be either in READY or STANDBY state. When exchanging traffic - transmitting or receiving data packets - the MS is in READY state. In this state, each cell change (cell reselection) is signalled to the network that tracks the MS at the cell level. Therefore, an arriving packets can be forwarded directly to the MS. The transition from READY to STANDBY is governed by the so-called "Ready_Timer" (denote by T_R), which is reset upon each packet transfer from / to the MS. When the MS is in STANDBY, i.e., after T_R seconds from the last packet, it does NOT signal the cell reselections to the network, unless the changes occurs to a different RA. Hence, the network can track the position of the MS in STANDBY only at the RA level. If a packet arrives directed to a MS in STANDBY, the network must perform a paging procedure in the whole RA in order to "discover" its current position, i.e., its current cell. The paging messages are generated by the SGSN: if the RA is covered by multiple BSC, the SGSN sends a paging message to each of them. The paging procedure consumes resources on the radio interface - the so-called Paging Channel (PAGCH) is dedicated to this purpose - and introduces an additional delay component on the arriving data packet, which is stored until the paging procedure is successfully completed. The value of the timer T_R is configurable by the operator to balance between the signaling due to cell reselection in READY state and the paging traffic in STANDBY. It is typically set in the range 0.5÷3 minutes. Similar mechanisms apply to UMTS, although a different terminology is adopted for the mobility

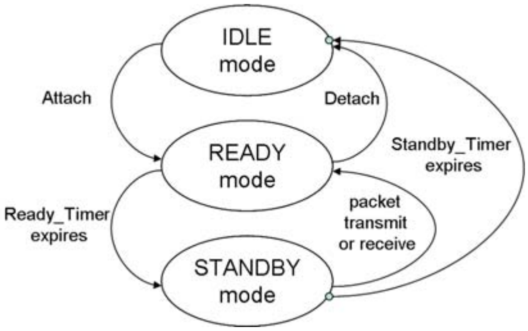


Fig. 9: GPRS mobility states.

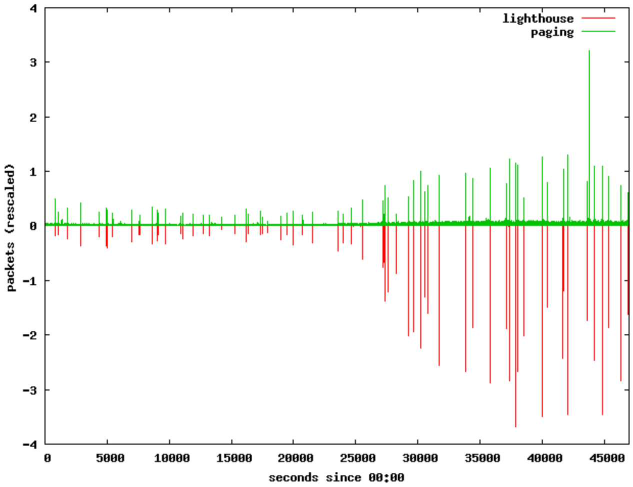


Fig. 10: Comparison between the *total* rate $A_s(t)$ of paging messages observed on the IuPS links (positive Y-axis) and the rate $G(t)$ of incoming scanning traffic observed on the Gn links (negative Y-axis) for a single UMTS SGSN, timebins of 1 sec.

states[2] and the transitions are somewhat more complex. For more details on the paging procedure and for the mobility states, refer to [3] or directly to the specifications [11, p. 123-126]. For the sake of simplicity, we stick to the GPRS terminology in the rest of this section.

Let $A_s(t)$ denote the rate at which the scanning probes arrive at the SGSNs at time t and let $G(t)$ denote the corresponding rate at which the paging messages are generated by the SGSNs. Furthermore, let α indicate the average fraction of active MSs that are in STANDBY state (assumed constant in time at first approximation) and $\overline{M}$ the average number of BSC/RNC covering each RA. With these positions, we obtain a simple relationship between the paging process at the SGSN and the scanning traffic:

$$G(t) \approx \alpha \cdot \overline{M} \cdot A_s(t). \qquad (1)$$

Recall that the intensity of the scanning traffic that reaches the SGSN $A_s(t)$ varies with time: since the GGSN drops the arriving packets directed to currently inactive addresses, $A_s(t)$ becomes proportional to the number of active MSs at time t. This follows the typical hour-of-day profile: lower in the night and maximum at the peak hour. Hence, the same hour-of-day profile modulates the paging process $G(t)$.

As noted above, at the time of measurements most of the incoming high-intensity scanning occurred on a single UDP port. Therefore the task of classifying scanning packets at the SGSN, i.e. measuring $G(t)$, could be reduced to simple port matching. On the other hand, the counting of paging messages required the complete monitoring of the IuPS and Gb links for two distinct SGSNs and the parsing of the signaling protocols on those interfaces (BSSGP and RANAP). Note that the process $A_s(t)$ represents the *total* paging rate, including that associated to legitimate non-scanning traffic. In the specific network under study, we found that during the measurement period, the overwhelming majority of the paging messages originating from the SGSN were indeed due to scanning traffic! This is evident from Fig. 10 and 11, reporting the measured values of $A_s(t)$ and $G(t)$ for two SGSNs (UMTS and GPRS), counted at 1 sec granularity[3] during the same half-day period starting from 00:00.

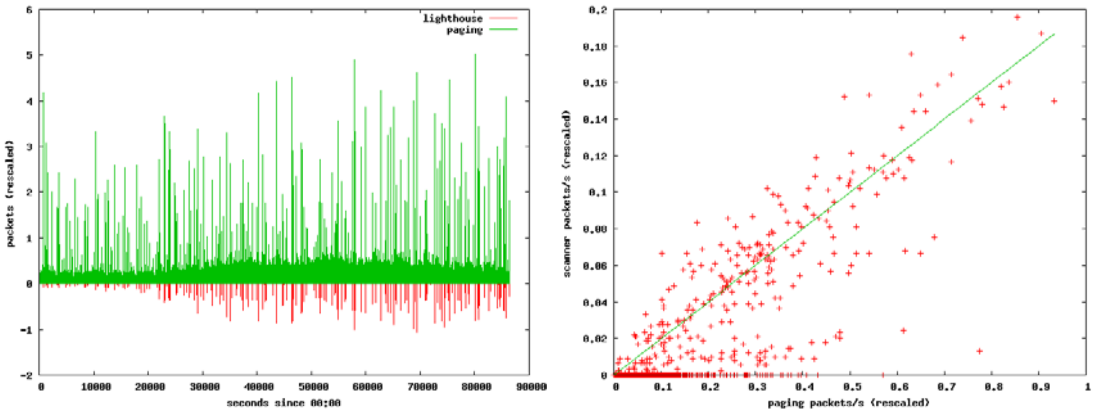
Fig. 10 reports the measurements for the UMTS SGSN. The paging process exhibits very high spikes clearly corresponding to the arrival of scanning bursts. The ratio between the spike sizes is approximately 1:3 (paging-to-scanning). In the specific network area under study, each UMTS RA is under the control of a single RNC, hence $\overline{M} = 1$ in eq. (1). Therefore, we conclude that approximately 30% of the active MS in UMTS are “pageable” (i.e., in STANDBY state) at any time. Notably, the residual background paging, i.e., *not* due to scanning spikes, remains pretty low: this is consistent with the fact that at present the most popular applications used in UMTS are based on client-initiated communications. In other words, the scanning process is by far the dominant cause of paging traffic in the UMTS domain.

The latter statement is also true for the GPRS domain, as can be seen from Fig. 11: here, the correspondence in time between the two time-series $A_s(t)$ and $G(t)$ (Fig. 11a) is less evident due to a higher level of variability in the residual paging traffic. In order to show the correlation between the two processes, we resort to the scatterplot of $A_s(t)$ vs. $G(t)$ in Fig.11b. The slope of the regression line shows that that the paging-to-scanning intensity ratio is now much higher, from 1:3 in UMTS up to approximately 5:1 in GPRS. The main reason is that in the GPRS domain RAs are commonly split among several BSC, hence $\overline{M} > 1$ in eq. (1).

These observations collectively demonstrate that scanning bursts from the Internet can penetrate into the Radio Access Network, causing bursts of paging messages in the packet-switched domain. This might in principle induce some small impairment for the users: in fact, if the rate of these paging spikes from the packet-switched domain is such as to congest the capacity available to the paging channel of the cell, the other paging messages originating from the circuit-switched domain (i.e., incoming voice calls) could be delayed or even dropped (unless the the latter are prioritized).

[2] The different terminology is prone to generate confusion. For example the IDLE state in UMTS corresponds to STANDBY in GPRS.

[3] Both graphs are rescaled by an undisclosed factor to avoid revealing the absolute number of active users.



(a) timeseries

(b) scatterplot

Fig. 11: Comparison between the *total* rate $G(t)$ of paging messages observed on the Gb links (positive Y-axis in (a), Y-axis in (b)) and the rate $A_s(t)$ of incoming scanning traffic observed on the Gn links (negative Y-axis in (a), X-axis in (b)) for a single GPRS SGSN, timebins of 1 sec.

VI. Related Works

The initial study on one-way delays in an operational IP network was performed by Papagiannaki et al. in [4] and was limited to a single-hop section, i.e., at a single router. In that study the methodology based on IP header hashing and matching was presented for the first time. The analysis was then extended to the end-to-end delays across a geographical network in [12]. These studies were performed on the Sprint backbone, while our study addresses an access mobile network. It is instructive to compare the findings of that study with our observations. In [4] they report occasionally large delay values caused by short-term interruption in the forwarding process: the so-called "coffee-break" events. These were considered as "normal" events in the router behaviour, decoupled from the arriving traffic pattern. We also found signs of coffee-breaks in our network, but unlike [4] we observed correlation with the activity of high-rate scanners. Similarly, in [12] the authors found sporadic large delay spikes associated with occasional high-rate bursts that were responsible for the rapid fill-up of some buffer. However, they did not provide any explanation regarding the source of these bursts.

An empirical analysis of the packet forwarding process in an operational router at very small timescales is given in [13]. There, the goal was to identify a plausible metric to summarize the traffic behavior at small-time scales, to be used for dimensioning. Our goal is different: our investigations of the delay process are ultimately aimed at characterizing the "physiological" delay process in the specific network section. The latter can be used as a reference baseline for detecting future drifts and/or anomalies that are symptoms of network internal problems.

Some previous works have studied the delay process in 3G mobile networks based on passive traces, e.g., [8], [14], [15]. They all focused on the analysis of Round-Trip-Times (RTT), estimated from TCP DATA/ACK or SYN/SYNACK pairs captured at a single monitoring point on Gi. Other works have resorted to active measurements to evaluate one-way end-to-end delays, e.g., [16], [17]. In general, it is not possible to exploit neither the RTTs nor the end-to-end delays to infer the internal dynamics of the Core Network. This is because the delay on the radio link is the dominant component in the total delay budget, and it exhibits a high degree of variability due to the changing radio conditions. Therefore, the fine-grain dynamics taking place within the wired section can not be observed with those approaches.

None of the previous studies reported on the impact of scanning traffic, but it is possible that some of the observations left unexplained there can be linked to the effects of scanning traffic. This hypothesis is consistent with the recent work [18] reporting that the "explosion" of scanning traffic can be dated back to 2001. Since then, it has remained a regular component of the so-called "unwanted traffic" (or "background traffic" [19]). The potential impact of such traffic on a 3G mobile network was first discussed in [20]. Here, we go one step further and analyze the actual impact in a concrete case-study.

VII. Discussion

The findings presented above are clearly specific to the particular network under study. Nevetheless, based on such collective results, it is possible to draw some more general lessons that we believe are useful for network engineers and researchers.

Large access networks employing public IP addressing are permeable to large bursts of probe packets generated by high-rate scanning sources. This has several consequences. First, it causes micro-congestion events on the links (buffer saturation) and in the nodes (CPU coffee-breaks). Regarding the latter, direct experimentation (i.e., stress-tests with synthetic traffic) would be needed to provide a definitive and accurate view of the impact of scanning burts on real routers and stateful elements (e.g. GGSN). This is ultimately a task for equipment testing rather than research. The lesson learned from this work is that scanning traffic is regularly present, and equipment vendors as well as network operators should be aware of the potential impact on their boxes. Second, micro-congestion events caused by scanning traffic might in turn trigger other hidden dynamics. An example is provided by the "packing" effect of uplink ACKs. There is no empirical evidence that the observed effects can cause any appreciable impairment to the quality experienced by the users and/or to the stability of the network, at least in the specific network under study. Nevertheless, gaining a better understanding of the dynamics at play is certainly a prerequisite for preventing potential problems that might emerge in the future or in different network configuration conditions.

Another aspect to be considered is that scanning traffic introduces a sort of "shot-noise" to the delay measurement process, i.e., large delay spikes due to micro-congestion events. This complicates the task of using the delay statistics (e.g. percentiles) to track the correct behaviour of the network elements and to detect misfunctioning and/or capacity shortage. A possible workaround would be simply to define statistical indicators that are insensitive to short-term delay spikes. Another approach would require to directly identify the scanning bursts and filter out the impaired samples from the overall statistics. In any case, we believe that the presence of such a "noise" in the measured delay process should be explicitly checked and taken into account in any experimental activity dealing with queuing analysis in real-networks. For example, there are a number of ongoing experiments aimed at filling the gap between theory and practice regarding buffer sizing in real networks (see [21] for a recent survey), and this phenomenon might be among the causes of mismatching between the expected and the measured delay statistics.

All the above findings apply to any IP-based access network. On the other hand, the observed (and expected) correlation between scanning traffic and the signaling process is highly specific to 3G cellular networks. We have shown that scanning traffic causes a considerable excess of paging traffic in the Radio Access Network, the consequences of which strongly depend on the detailed configuration of the network (e.g., setting and capacity of paging channels). This is yet another example of how "unwanted" traffic can generate wastage of physical and logical resources in the radio network, an issue already raised [20], and that we believe is worth further attention by the research community.

By a curious paradox, a collective view of the findings reported in this case-study show that sequential scanning traffic can also play a *positive* role for the network operator: thanks to its "extreme" characteristics it *evokes hidden dynamics* so as to expose latent weaknesses - e.g., routing loop, bottleneck link, CPU coffee-breaks, permeability of the paging process, only to stay with the examples found in this case-study - which can then be observed and fixed by the network expert. In other words, regular scanning traffic can be exploited for "opportunistic detection" of network problems if coupled with accurate traffic monitoring.

It is important to remark that the observed scanning traffic was produced by external Internet hosts which *were not targeting our network* specifically. On the other hand, it is evident that properly crafted scanning patterns might be used to launch deliberate Denial-of-Service (DoS) attacks against the cellular network infrastructure. For instance, the paging spikes due to scanning bursts reported in §V recall immediately the "paging overload" attack model presented in a recent work [22]. Further types of attack models can be designed that exploit the same dynamics evoked by unwanted traffic (see [10]), and particularly scanning traffic. In other words, there is a duality between the impact of "background" unwanted traffic and deliberate attack models. On the positive side, this implies that the "opportunistic detection" of certain critical dynamics evoked by unwanted traffic also provides a precious insight into new potential attack models. We believe this is an important aspect that is worth further attention by the research community.

VIII. Conclusions and Future Work

In this work, we have provided a number of empirical observations from a real operational network, specifically the core section of a large 3G mobile network. We found that the incoming traffic process from the Internet yields positive impulses (spikes) and investigated their causes. The main discovery is that Internet traffic contains large bursts of packets generated by sequential scanning activities at high rates. Cellular 3G networks employing public IP addressing are permeable to such traffic. This has a number of consequences that we have started to unveil.

A more general question remains on the ground: Should scanning traffic be permitted to enter the network ? We do not see any productive application of sequential scanning from the perspective of the users. On the other hand, blocking such traffic without impairing legitimate applications is a non trivial task (see e.g. [23] [24]).

The primary lesson to be learned from this work is that the accurate analysis of passive traces is a powerful means in support of network operation and preventive troubleshooting. The findings reported here collectively represent an illustrative case-study from a real operational network: it is remarkable how much accurate information one can extract out of the exploration of a few packet-level traces. This information has often direct implications for the operation and management of the network itself. We have shown that the microscopic analysis of one-way delays plays an important role in this approach. We are now seeking to implement delay measurements aquisition, processing and reporting as an automatic on-line feature for our monitoring system, to be adopted on a production basis in support of network operation. The ultimate goal is to implement an automatic tool that is able to detect changes and/or drifts in the packet delay process at different time-scales on selected network sections. We believe this to be a cost-effective strategy to detect events like equipment misfunctioning, shortage of capacity, and emergence of critical traffic patterns.

Acknowledgments

This work was supported by the *Kplus* programme through the DARWIN project [2]. The views expressed in this paper are those of the authors and do not necessarily reflect the views of the funding partners. We thanks Jossy Sayir for his help with the final editing.

REFERENCES

[1] Endace Measurememt Systems. [Online]. Available: http://www.endace.com

[2] METAWIN and DARWIN projects. [Online]. Available: http://userver.ftw.at/ ricciato/darwin

[3] J. Bannister, P. Mather, S. Coope, *Convergence Technologies for 3G Networks*. Wiley, 2004.

[4] K. Papagiannaki, S. Moon, C. Fraleigh, P. Thiran, C. Diot, "Measurement and Analysis of Single-Hop Delay on an IP Backbone Network," *IEEE JSAC*, vol. 21, no. 6, August 2003.

[5] "The MD5 Message-Digest Algorithm," *RFC 1321*, April 1992.

[6] P. Svoboda et al., "Composition of GPRS/UMTS traffic: snapshots from a live network," *4th Int'l Workshop on Internet Performance, Simulation, Monitoring and Measurement (IPS-MOME'06), Salzburg*, February 2006.

[7] L. Zhang, S. Shenker, D. Clark, "Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two-Way Traffic," *ACM Computer Communication Review*, vol. 21, no. 4, September 1991.

[8] F. Vacirca, F. Ricciato, R. Pilz, "Large-Scale RTT Measurements from an Operational UMTS/GPRS Network," *1st Int'l Conference on Wireless Internet (WICON'05), Budapest*, July 2005.

[9] U. Hengarthner, S. Moon, R. Mortier, C. Diot, "Detection and Analysis of Routing Loops in Packet Traces," *2nd ACM SIGCOMM Workshop on Internet Measurement (IMW'02), Marseille*, 2002.

[10] F. Ricciato, P. Svoboda, E. Hasenleithner, W. Fleischer, "On the Impact of Unwanted Traffic onto a 3G Network," *Proc. of 2nd Int. workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPeru'06), Lyon*, June 2006, (an earlier version is available as Technical Report FTW-TR-2006-006 from [2]).

[11] "Universal Mobile Telecommunications System (UMTS); General Packet Radio Service (GPRS); Service description; Stage 2," *3GPP TS 23.060, Version 7.5.0, Release 7*, October 2007.

[12] B. Choi, S. Moon, Z. Zhang, K. Papagiannaki, C. Diot, "Analysis of Point-To-Point Packet Delay in an Operational Network," *IEEE INFOCOM'04, Hong Kong*, March 2004.

[13] K. Papagiannaki, R. Cruz, C. Diot, "Network Performance Monitoring at Small Time Scales," *ACM Internet Measurement Conference (IMC'03), Miami*, 2003.

[14] P. Benko, G. Malicsko, A. Veres, "A Large-scale Passive Analysis of End-to-End TCP Performance over GPRS," *IEEE INFOCOM'04, Hong Kong*, March 2004.

[15] J. Kilpi, P. Lassila, "Micro- and macroscopic analysis of RTT variabilityin GPRS and UMTS network," *Proc. of Networking 2006, LNCS 3976, Coimbra, Portugal*, May 2006.

[16] Y. Lee, "Measured TCP Performance in CDMA 1x EV-DO Network," *Proc. of 7th Passive and Active Measurement conference (PAM 2006), Adelaide, Australia*, March 2006.

[17] J. M. Cano-Garcia, E. Gonzalez-Parada, E. Casilari, "Experimental Analysis and Characterization of Packet Delay in UMTS Networks," *Proc. of 6th Int'l Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking, St. Petersburg*, May 2006.

[18] M. Allman, V. Pasxson, J. Terrell, "A Brief History of Scanning," *ACM Internet Measurement Conference (IMC'07), San Diego*, October 2007.

[19] R. Pang et al., "Characteristics of Internet Background Radiation," *ACM Internet Measurement Conference (IMC'04), Taormina*, October 2004.

[20] F. Ricciato, "Unwanted Traffic in 3G Networks," *ACM Computer Communication Review*, vol. 36, no. 2, April 2006.

[21] Y. Ganjali, N. McKeown, "Update on Buffer Sizing in Internet Routers," *ACM Computer Communication Review*, vol. 36, no. 5, October 2006.

[22] J. Serror, H. Zang, J. C. Bolot, "Impact of paging channel overloads or attacks on a cellular network," *Proc. of 5th ACM workshop on Wireless security (WiSe'06), Los Angeles*, September 2006.

[23] J. Twycross, M. M. Williamson, "Implementing and testing a virus throttle," *Tech. Report HPL-2003-103*, May 2003. [Online]. Available: www.hpl.hp.com/techreports/2003

[24] V. Falletta, F. Ricciato, "Detecting Scanners: Empirical Assessment on a 3G Network," *International Journal of Network Security (submitted)*.

Fabio Ricciato received a the Laurea degree in Electrical Engineering and the PhD from the University La Sapienza in Rome, Italy. In 2004 he joined the Telecommunications Research Center of Vienna (ftw.) as Senior Researcher and Project Manager for the METAWIN and DARWIN projects. He is now Assitant Professor at the University of Salento, Italy, where he teaches a course on Telcommunication Systems. He collaborates with ftw. where recently was appointed scientific Area Manager for the Packet Networking department.

Eduard Hasenleithner received a DI(FH) degree in telecommunications engineering from the Salzburg University of Applied Sciences, Austria in 2002. Since 2001 he is with the Telecommunications Research Center Vienna, working on different networking topics with focus on experimental and laboratory activity. He is one of the main developers of the METAWIN monitoring system and a staff member the DARWIN project.

Peter Romirer-Maierhofer received a DI(FH) degree in telecommunications engineering from the Salzburg University of Applied Sciences, Austria, and a M.Sc. degree in computer systems engineering from the Halmstad University, Sweden, in 2005. He is a member of the DARWIN project on traffic analysis in GPRS/UMTS networks at the Telecommunications Research Center Vienna. Currently, he is working towards the Ph.D. degree in computer science in the field of anomaly detection in 3G cellular networks.