Trustworthy Operations in Cellular Networks: The case of PF Scheduler

Konstantinos Pelechrinis, *Member, IEEE,* Prashant Krishanmurthy, *Member, IEEE,* and Christos Gkantsidis, *Member, IEEE*

Abstract—Cellular data networks are proliferating to address the need for ubiquitous connectivity. To cope with the increasing number of subscribers and with the spatio-temporal variations of the wireless signals, current cellular networks use opportunistic schedulers, such as the Proportional Fairness scheduler (PF), to maximize network throughput while maintaining fairness among users. Such scheduling decisions are based on channel quality metrics and Automatic Repeat reQuest (ARQ) feedback reports provided by the User's Equipment (UE). Implicit in current networks is the *a priori* trust on every UE's feedback. Malicious UEs can thus exploit this trust to disrupt service by intelligently faking their reports. This work proposes a trustworthy version of the PF scheduler (called TPF) to mitigate the effects of such Denial-of-Service (DoS) attacks. In brief, based on the channel quality reported by the UE, we assign a probability to possible ARQ feedbacks. We then use the probability associated with the actual ARQ report to assess the UE's reporting trustworthiness. We adapt the scheduling mechanism to give higher priority to more trusted users. Our evaluations show that TPF (i) does not induce any performance degradation under benign settings, and (ii) it completely mitigates the effects of the activity of malicious UEs. In particular, while colluding attackers can obtain up to 77% of the time slots with the most sophisticated attack, TPF is able to contain this percentage to as low as 6%.

Index Terms—Cellular networks, PF scheduler, Trust, Misreporting attack

1 Introduction

The dominant traffic in cellular networks has recently transformed from voice to data with the emergence of smartphones and improved network capabilities (e.g., transmission rates of up to 14 Mbps). The increasing importance of cellular data networks makes them an attractive target for Denial of Service – DoS – attacks (for a description of attacks reported in real networks, please see [1]). The vulnerabilities of cellular networks to a number of novel DoS attacks differ from the threats in other technologies and may be facilitated by the proportional fair (PF) scheduling mechanism employed over the downlink in cellular networks. The operations of the PF scheduler are based on the presumably honest feedback reports received from mobile stations, also called User Equipments (UEs). The two feedback reports of interest are (i) the channel quality indicator (CQI) and (ii) the transmission outcome (ACK/NACK). A malicious UE can cleverly manipulate his reports to disrupt regular network operations and gain unfair medium access, thereby causing starvation to legitimate users. We refer to these types of threats as **misreporting attacks**.

Cellular network protocols have mainly focused on performance, ignoring security implications. For instance, in the case of the scheduling algorithm, the base station completely trusts the reports from the UEs in order to achieve its objective (i.e., maximize network throughput, while maintaining fairness among the users). Misreporting attackers take advantage of this *a priori* trust in order to disrupt normal network operations. Previous studies have mainly focused on a single type of feedback; either erroneous channel quality feedback (e.g., [2]), or misreports of the packet transmission success (e.g., [3]). In our study, we consider the impact of both feedback reports. We propose a variation of PF scheduler that makes use of quantification of normal behavior to estimate and incorporate the *reporting trust* of each user. We will refer to our scheme as Trustworthy Proportional Fairness (TPF) scheduling. We show that TPF can efficiently mitigate the impact of misreporting attacks.

In brief, TPF works as follows. It assesses the *reporting trust* on a UE (say Jack's) by using (i) the expected outcome of a packet transmission (based on the reported channel quality indicator), and (ii) the reported feedback on the transmission outcome (e.g., ACK/NACK). When Jack reports the downlink quality he observes to the base station (BS), the BS decides upon a transmission rate to send data to him. This decision is taken on the basis of a predefined success probability p_s (typically 90%) of a frame at this rate. In other words, every packet transmission should be typically followed by an ACK with probability p_s, provided Jack's reports are credible. If a NACK is reported, this could have happened with a probability of $1 - p_s$, under the same assumption, and thus TPF keeps this as a measure of the trust on Jack's report. The BS accumulates observations and processes them to enable TPF to first update its esti-

- *Konstantinos Pelechrinis and Prashant Krishnamurthy are with the School of Information Sciences at the University of Pittsburgh.*
- *Christos Gkantsidis is with Microsoft Research at Cambridge, UK.*

Digital Object Indentifier 10.1109/TPDS.2013.144

mate on Jack's (reporting) trust and then his scheduling priority accordingly. Given the unpredictable nature of wireless medium, it is clear that a transmission can fail for legitimate users (say Bob) as well. This can have an impact on the perceived trust of Bob from the network's point of view, and consequently a negative effect on his share of the medium. Nevertheless, our evaluations indicate that under benign settings TPF performs as well as PF.

This work makes three contributions:

- We quantify the loss of network performance due to feedback misreporting.
- We present a method for assessing the trustworthiness of UEs.
- We show how this trustworthiness can be used to improve the PF scheduler's robustness against misreporting attacks.

In a nutshell, the misreporting attackers can gain up to 2.5 times more time slots compared to their fair share. However, TPF is able to restore fair network operations, and even almost nullify the time share of malicious entities.

Scope of our work: Research in wireless networking and security has followed seemingly disjoint paths. Wireless network protocol design has focused on performance, ignoring to some extent security implications. To secure today's wireless networks, some of the vulnerabilities that exist will have to be addressed, possibly with patchwork solutions. Going forward, a holistic view of security and performance is needed with a new paradigm of thinking in designing protocols and architectures. Our study clearly works towards this direction, providing an example of a protocol design that achieves the desired performance characteristics, while at the same time being robust against malicious behavior.

Furthermore, we would like to emphasize on the fact that even though there are no reports to date of large scale misreporting attacks, we act proactively. We identify the threat of a priori trusting the UEs and we provide a framework for assessing UEs reliability. Moreover, the attack model we consider does not require *hacking* of the cellular network, but at a device level. The open (software/hardware) nature of mobile handheld devices, make it even more possible that similar threats can be actually realized in the near future [2]. Note here that, our trust framework can be also applicable to cellular network's functionalities other than scheduling.

A preliminary version of this work can be found in [4]. Compared to [4] we have included in this manuscript additional evalution results and a detailed description of our customized, measurement-driven simulator (parts of which are provided in the Supplementary Material). The rest of the paper is organized as follows. Section 2 provides the required background on 3G cellular networks, discusses related studies, and differentiates our work from the existing literature. Section 3 details the threat model of misreporting attackers. Section 4 presents our reporting trust estimation module and its integration with the PF scheduler. Section 5 starts by briefly presenting our simulator. It further quantifies the effect of the different types of reporting misbehaviors and presents the evaluation of our proposed scheme. Finally, Section 6 concludes our work.

2 Background and Related Work

This section provides a brief background of 3G cellular networks and discusses other studies related to our work. We further differentiate our work from the existing literature. We provide additional details on cellular network operations in the Supplementary Material.

2.1 Cellular Networks

The basic topology of a cellular network consists of a base station (also called Node B) that serves all UEs that are within its coverage area. In our work, we consider the downlink with High Speed Packet Access (HSPA), which currently supports transmission rates of up to 14.4 Mbps[1]. In the following we describe four important mechanisms of interest in the downlink transmission protocol, HSDPA (High Speed Downlink Packet Access) of HSPA.

PF scheduler: The allocation of downlink capacity among the clients of a base station (BS) follows a Time Division Multiplexing (TDM) fashion, using the Proportional Fairness scheduling scheme ([6]) in time slots (Transmission Time Intervals - TTIs) of 2ms. Each user is assigned a *priority value* $p_i(t) = \frac{CQI_i(t)}{A_i(t-1)}$ that depends on the Channel Quality Indicator $CQI_i(t)$, a measure of the instantaneous downlink quality of user i, and $A_i(t-1)$, the average throughput that has been obtained by UE i until time slot $t-1$. $CQI_i(t)$ is reported by the UE to the BS and dictates the sustainable downlink rate in the current time slot. It takes values between 0 and 30, with 0 indicating no connectivity and 30 implying very good connectivity. Each UE estimates the link quality through pilot bits sent from the BS on the Common Pilot Channel (CPICH)[2]. The throughput value $A_i(t)$ is maintained by the network. Whenever user i sends an ACK, the BS increases $A_i(t)$. When a NACK is received, $A_i(t)$ is not updated, except when the retransmission limit (L_{max}) is reached. In this latter case, the BS increases $A_i(t)$ to reduce the priority of UE i and free the medium for other UEs. The UE with the highest priority $p_i(t)$ is scheduled during TTI t. Thus, PF approximately schedules the UE that has *the best link quality* to the BS and has been *served the least* during the previous time slots.

Hybrid Automatic Repeat reQuest (HARQ): As mentioned before, the channel quality determines the downlink transmission rate. The rate is chosen so as to achieve a target success probability (typically 90%) in a

1. We consider 3GPP Release 6 [5]. Note that later standards that make use of higher-level modulation schemes and MIMO can achieve even higher rates.

2. CQI reports happen either periodically through an HSPA Control Channel or are piggybacked in ACK/NACKs.

TTI. If the transmission is successful, an honest UE will send an ACK to the BS, while if the transmission fails, a NACK will be sent and retransmission is performed in the next time slot (fast retransmit mode)[3]. HSPA uses HARQ for the retransmission, which is a combination of forward error correction and error detection. In a nutshell, the erroneous packets are not discarded from the UE, but they are buffered and are *soft combined* with subsequent retransmissions [7]. As a result, there is an increased probability of success for retransmissions. Table 1 presents the corresponding packet success probability after a different number of HARQ transmissions [8].

# of HARQ transmissions	1	2	3
Success probability	0.791	0.905	0.938

TABLE 1
HARQ performance [8]

Rate selection: Based on the reported CQI, the BS selects the Modulation Coding Scheme (MCS) and the amount of data (Transmission Block Size - TBS) to transmit to the UE. In practice, the mapping from CQI to MCS and TBS depends also on the capabilities of the user terminal (see [9] for more details). For brevity, we shall ignore the existence of different categories of UEs, and focus on the fact that CQI determines the transmission rate, through the choice of MCS and TBS. We would like to emphasize on the fact that since the TTI is fixed, it might be possible that application layer packets will be fragmented into many smaller transmission blocks. For the rest of the paper, we will use the terms packet and TBS interchangeably.

2.2 Related Studies

Racic *et al.* study the effect of fake CQI reports [2]. They consider both single and colluding attackers, as well as intra- and inter-cell attacks. They argue that inter-cell attacks are more effective. Once the attacker UE has reached its maximum possible CQI report, it can handoff to a new BS, report an arbitrarily low throughput A_i and be scheduled again. The authors also propose and evaluate a robust handoff scheme. However, they do not consider the actual architecture of a 3G network, where handoffs are handled by the Radio Network Controller (RNC) and thus, the values of A_i can be verified across BSs. In current networks, the RNC centralizes many network functionalities and "transforms" a multi-cell topology to a (large) single cell, for the operations that we consider in this work. Therefore, we shall ignore inter-cell attacks, and focus on intra-cell attacks.

In another type of fake report attacks, Ben-Porat *et al.* study the retransmission attack [3]. A malicious UE can increase its time share if it persistently reports NACKs whenever it is scheduled. After showing the effect of such behavior, the authors propose a way to update the A_i values after each transmission to UE i, which is immune to retransmission attacks and retains fairness. Nevertheless, the applicability of their scheme is limited to the specific type of attack. Kim and Hu [10] study the problem of fake CQI reports as well, and propose a challenge-response scheme to prevent manipulated feedback. In brief, the BS transmits "challenges" to the UE containing a random (known to the BS) pattern at different rates. The UE, when it correctly receives the challenge, reports back to the BS the obtained pattern and thus the real sustainable rate is revealed. The challenges are created in such a way that the attacker cannot guess them. The system is shown to effectively thwart CQI misreporting attacks. However, note here that the scope of the proposed solution is limited (e.g., it cannot deal with retransmission feedback attacks).

Even though not tightly related with the feedback misreporting attacks, Bali *et al.* ([11]) identify a vulnerability of the PF scheduler related to the traffic pattern of the UEs. In particular, they experimentally show that the PF scheduler is sensitive to downlink *on-off* traffic, that is, "periodic", non-backlogged traffic for a specific UE. In their experiments, they consider a downlink burst/stream of 250 packets of 1500 bytes every 6 seconds for the malicious UE and a long-lived downlink UDP packet stream for the well-behaved UE with an average rate of 600Kbps. The network does not distinguish between UEs that are backlogged and UEs that do not have pending traffic at the BS and treats them in the same way. Hence, a user i that currently does not have any downlink traffic, will have its A_i reduced, since he does not obtain any data from the BS. Consequently, i will have an *inflated* priority value p_i next time there is downlink traffic for it at the BS. This will cause starvation to the rest of the users, since i will obtain consecutive time slots until its p_i is reduced. However, the authors experiment in a limited setting with only 2 UEs and they do not provide results for the network wide effects of the *on-off* traffic. As identified by the authors, many (well behaved) UEs can have similar traffic patterns and the presence of many such UEs may tone down this effect.

To the best of our knowledge, *we are the first to incorporate a UE trust module in the PF scheduler. Unlike other solutions proposed, TPF is able to mitigate the impact of a broad class of attacks related to malicious misreporting from the UE*. We also delve into the detailed effects of the different types of misreports.

3 Misbehavior Model

Current cellular protocols assume that UEs are honest and cooperate for the *optimal* operation of the network. In particular, they assume that UEs provide accurate feedback about the channel quality they observe and the successful reception of the transmitted frames. However,

3. It is possible to have slow retransmit mode, where the retransmission is performed the next time the user is chosen to be served from the PF scheduler.

malicious users can manipulate this feedback, by faking CQI and ARQ reports, leading to different levels of performance degradation for legitimate users. In the following, we delve into the effects of each misreport. We also consider selfish users, who employ the same malicious techniques, but with the primary goal of increasing their throughput (rather than simply causing starvation to the rest of the users). Our discussion in this section is qualitative; Section 5 quantifies the impact of misreports.

3.1 Fabricated CQI

The link quality reported from Jack's UE affects directly the numerator of $p_{Jack}(t)$, $\mathrm{CQI}_i(t)$. By increasing this value, Jack increases his priority and his share of the medium (compared to legitimate users). However, a higher CQI value corresponds to a larger packet size (TBS). This translates to an increase in the denominator of p_{Jack}, $A_i(t-1)$, either when the retransmission limit is reached (i.e., all packets are received with error, or Jack misreports ACKs) or when an ACK is being sent back to the BS (e.g., Jack truthfully reports a successful packet reception). This potentially reduces Jack's priority, *moderating* the effect of the increased CQI report.

Note here that, for a selfish user whose goal is to increase his throughput and hence receive the transmitted data correctly, increasing CQI reports drastically reduces the chances of decoding the packet successfully. Recall Table 1 that presents the decoding probabilities under the assumption of correct CQI reports. When the difference $\Delta CQI = CQI_{reported} - CQI_{actual}$ is not zero, we obtain the results presented in Table 2 [8]. As we can see from these data, when Δ_{CQI} is 1, the probability that the first transmission succeeds is only 3% (as compared to the 80% probability with the correct CQI report). This probability increases in the subsequent retransmissions due to the HARQ, however it is still very small (5% and 7%). Note here that when the CQI reported is 2 or more units higher than its actual value, the probability of successful reception is practically 0 (even after 2 retransmissions). The intuition behind this, is as follows. Higher CQI reports from the UE will lead the BS to transmit at a higher rate, since it assumes that the UE supports that rate. However, since this is not true, the packet will not be decoded correctly with high probability. We will come back to this issue in our simulation results in Section 5.2.

# of HARQ transmissions	$\Delta CQI = 1$	$\Delta CQI > 1$
1	0.03	0
2	0.05	0
3	0.07	0

TABLE 2
Success probability for fabricated CQI reports [8]

Operations: Our fabricated CQI attack model assumes the best case scenario for the attacker(s), that is, Jack has *complete knowledge* of the priority values of the legitimate users. Let us first consider the case where Jack is the only misbehaving user. (a) When Jack wants to cause DoS he can estimate the minimum possible CQI value that will render his priority the maximum. Depending on the link qualities of the legitimate users, it is possible that the optimal CQI value is lower than the actual one[4]. Jack will report the smaller value in order to deter the BS from using higher transmission rates and largest packet sizes. In doing so, he delays the increase in $A_{Jack}(t-1)$, and, hence, he can extend the duration of his attack. Eventually, he will need to inflate the reported value of CQI. (b) When Jack acts selfishly, he reports the minimum value that will render his priority the maximum with the constraint that this value is at least as high as his actual CQI. The reason for imposing this lower bound on his reported CQI is the fact that the ultimate goal of Jack is to actually increase his throughput, not cause DoS. Reporting a CQI value which is lower than his actual one will eventually lead to a lower throughput.

We also consider multiple misbehaving nodes. In such cases, we will assume that they collude. There are many different ways for them to cooperate, but Racic *et al.* ([2]) have shown that the most effective colluding strategy is the **Delta CQI attack**. In a Delta CQI attack, every attacker (say Bob) calculates the increase, δ_{Bob}, of his CQI value needed in order to get a higher priority as compared to the legitimate users. The user who has the smallest value of δ is the one who reports the fabricated CQI. Again, note that for selfish behavior the value of δ cannot be negative.

We acknowledge that in reality, it is not easy (if possible at all) for a malicious UE to know all of the network parameters (e.g., other UEs' priority values etc.). Nevertheless, by making this assumption, we show that our approach is able to deal even with the most advanced attack models. On the contrary, an attacker without this knowledge can potentially have a lot less effect on the network operations. Racic *et al.* [2] have shown that if it constantly reports the maximum possible CQI, this will reduce its priority value very fast, reducing at the same time the number of time slots obtained from the malicious UE. Therefore, the attacker will need to follow an exploratory approach with regards to the CQI value that it reports. For instance, reporting random CQI values, possibly chosen from a distribution biased towards high CQI values, can only have an effect in the case where legitimate users have really poor link qualities to the BS and for a limited amount of time. Racic *et al.* [2] have also provided an approach for the attacker to estimate/approximate the "optimal" CQI value to be reported for its attack. Collusion can further increase the attack effectiveness, but again it will depend on the number of attackers as well as the attack strategy

4. Of course as the number of UEs covered by a BS increases, the chances of obtaining the medium with a lower CQI drastically decrease.

followed. Nevertheless, even in this case, TPF will still be able to restore a significant portion of the "malicious" time slots to the honest users (as we will see in Section 5.2) by reducing the trust value of an attacker very fast.

3.2 Fabricated ARQ feedback

In addition to CQI manipulation, Jack can fabricate his ARQ feedback report. This can happen independently of CQI misreports. By constantly reporting a failed reception (i.e., NACK), Jack forces the BS to perform retransmissions, starving the rest of the users. Note here that, it can be the case that a NACK is the correct feedback (e.g., when Jack's UE reports a higher CQI than what he actually has, i.e., $\Delta CQI > 1$). However, the thesis here is that Jack's UE does not consider at all the result of the decoding but constantly reports a NACK.[5] It should be apparent that the ARQ feedback is the major factor that dictates the severity of the misbehavior. An increased number of NACKs leads to higher levels of starvation for legitimate users. As one might expect from the above discussion, reporting a fake ARQ feedback without fabricated increased CQI reports leads to a larger degree of starvation as compared to the case where both reports are fabricated (more details are provided in Section 5).

Operations: A malicious UE mainly reports negative acknowledgements in order to starve legitimate users. Each UE is associated with an *a priori* trust value which controls the portion of NACKs being sent back to the BS. In other words, if the a priori trust of Jack's UE is $k \in [0, 1]$, he will transmit a NACK with probability $1-k$ for every downlink (re-)transmission.

Table 3 summarizes the possible combinations of fake or correct reports. We also preview a qualitative description of the severity of each one of the combinations, leaving their quantitative evaluation for Section 5. Note here that, even though a selfish user cannot really increase his throughput significantly, his behavior has a big effect on the rest of the users.

# CQI report	ARQ report	Behavior	Effect level
✓	✓	Benign	No effect
✓	✗	DoS	Severe
✗	✓	Selfish	Big
✗	✗	DoS	Severe

TABLE 3
Different Combinations of Misreports and Their Effect

Summary of attack models: To sum up, we consider a colluding attack model, where malicious UEs have full knowledge of the priority values of the legitimate users. For the fabricated CQI attack, misbehaving users collude using the Delta CQI strategy (as described above). For the fabricated ARQ feedback no collusion is required. Attackers can deploy their misreporting strategy either constantly or probabilistically. In the latter case, the probability of attack is constant across every downlink transmission (i.e., it is not time varying).

5. In the following we will also consider cases where Jack probabilistically decides to report an ACK in order to *confuse* the BS.

4 Truthful Scheduling

In this section we begin by presenting our scheme for assessing the reporting trust of a UE. Later we show the integration of the estimation module with the scheduling mechanism, towards the trustworthy proportional fair scheduler.

4.1 Reporting Trust Assessment

The terms "trust" and "trustworthiness" can be defined in many ways [12] and are typically context dependent. In our work, the trust level of a user's equipment, say Jack's, is associated with its reliability and correct implementation of the network functionalities.[6] Formally, the trustworthiness of a UE is the probability that it correctly performs the operations mandated by the network. In this paper, we focus on the correct reporting of the CQI and ARQ values, which directly affect the scheduling decisions of the network. In other words, we seek to answer the following question: *"What is the probability that Jack's UE is reporting the correct CQI and ARQ?"*

Our approach in a nutshell: To assess the trust on Jack's UE, the BS monitors all feedback received from Jack's UE. Every ARQ feedback (ACK/NACK) following a downlink transmission to Jack's UE, is considered as an *observation*. Based on a set of k observations, the BS statistically estimates the probability that Jack's UE has correctly reported both CQI and ARQ feedbacks, using a Maximum Likelihood Estimation (MLE) framework. During this process we take into consideration wireless induced effects, thus capturing the probability that an observation might be negative due to link failures and not due to manipulated reports from Jack's UE.

Trust representation: A strict notion of trust could be represented by a binary variable Y, which would be 0 if the node is (always) untrustworthy and 1 otherwise. Nevertheless, in reality not only may a UE act in between the two extremes, but there is also uncertainty in establishing trust based on observations. Therefore, we consider Y to be the likelihood that the UE is trustworthy and as such it is a real number in the interval [0, 1]. We assume that the network initially completely trusts every UE. These (initial) trust values dynamically evolve as UEs interact with the BS. For an untrustworthy UE its trust will *eventually* converge to a low value.

As alluded to above, we use the ARQ feedbacks as observations to establish the trustworthiness of Jack's UE. Let us denote by o_i, the *reported* feedback, with $o_i = 0$ when a NACK is received (i.e., Jack's UE reports

6. For our purposes it is not important to distinguish whether the UE has been compromised or whether its owner is malicious. We ignore the reasons for malicious behavior, and simple refer to the UE's and user's trust interchangeably.

a packet transmission failure), and $o_i = 1$ otherwise. The outcome of the interactions depends on (i) the wireless link quality and (ii) reporting trust on Jack's UE.

The BS monitors the outcome of k consecutive packet transmissions to Jack. These observations form a sample set, indexed by j. For each packet transmission, i, BS records o_i and the probability that the packet must have been successfully transmitted on the downlink $p_{s,i}$ (s is the number of retransmission counts for the packet). The latter is estimated from the reported CQI values; recall that the success probability increases with each retransmission, i.e., success is higher for higher values of s (see Table 1).

Let us now assume that the BS associates a trust value $t_{i,Jack}$ (with Jack's UE) during the i^{th} packet transmission. Then it is easy to see that the i^{th} packet transmission is a Bernoulli trial Z, with probability of success $p_{s,i} \cdot t_{i,Jack}$. Thus, the probability density function (pdf) of Z is:

$$f_i(X = o_i) = (p_{s,i} \cdot t_{i,Jack})^{o_i} \cdot (1 - p_{s,i} \cdot t_{i,Jack})^{1-o_i} \quad (1)$$

The base station assumes correct values for the reported CQI when computing the probabilities $p_{s,i}$ (correct both in terms of accurate channel conditions estimation by the hardware as well as truthful reporting of the CQI by Jack's UE). We shall also assume that during the k transactions forming the j^{th} sample set, the reporting trust of Jack's UE as perceived from the BS is constant[7], i.e., $t_{i,Jack} = t_{Jack}, \forall i \in \{1, 2, ..., k\}$. We will next propose a maximum likelihood estimation (MLE) method to update our estimate of t_{Jack} utilizing the k observations.

MLE is a statistical method that estimates the parameters of a distribution based on a set of observations. Let us assume that we have a set of observations $\overrightarrow{o}$, that are drawn from a parametric distribution with density $f_{\overrightarrow{p}}(\cdot)$, with $\overrightarrow{p}$ being the parameter vector of the distribution. If $\overrightarrow{p}$ is unknown, $f_{\overrightarrow{p}}(\overrightarrow{o})$, is called *likelihood function*. In the following we will use the more convenient, *log-likelihood function*, $log(f_{\overrightarrow{p}}(\overrightarrow{o}))$.

MLE computes the vector $\overrightarrow{p}$, based on the k observations forming vector $\overrightarrow{o}$. In brief, MLE estimates the parameter vector, such that the log-likelihood function is maximized for the observed vector $\overrightarrow{o}$. Formally, $\overrightarrow{p}$ is estimated as the solution of the following optimization problem:

$$\text{maximize} \qquad \frac{1}{k} \cdot \log(\prod_{i=1}^{k} f(o_i|\overrightarrow{p})) \quad (2)$$

$$\text{subject to} \qquad \overrightarrow{p} \in C \quad (3)$$

where C is the domain set of the parameter vector. Intuitively, MLE computes a value of $\overrightarrow{p}$ that maximizes the likelihood that the set of k observations were indeed the outcomes of k independent experiments. The event that all the outcomes are jointly obtained translates to an "AND" operation. This likelihood of the joint event is a product of the conditional probability density functions; maximizing this is equivalent to maximizing the sum of the logs of the conditional density functions. Note here that MLE can also deal with heterogeneous data; observations are drawn from the same distribution family but some, known, parameter of this family can be different across the different samples. The interested reader can find more information on MLE techniques in [13].

Thus, returning back to the trust assessment problem, the BS's trust on Jack's UE is obtained as the solution to the optimization problem:

$$\max_{t^j_{Jack}} \qquad \frac{1}{k} \cdot \sum_{i=1}^{k} \log(f_i(o_i|t^j_{Jack})) \quad (4)$$

$$t^j_{Jack} \in [\widehat{t_{Jack}}, 1] \quad (5)$$

where t^j_{Jack} is the estimate of Jack's trust, based on sample set j. Given $\overrightarrow{o^j_j}$, the trust in Jack cannot be smaller than the percentage of successful transactions in $\overrightarrow{o^j_j}$. When $\overrightarrow{o^j_j} = \oslash$, $\widehat{t_{Jack}}$ captures the non-zero probability that all packets sent to Jack in the sample window are dropped because of wireless induced failures. This probability is tied to the event of having no knowledge about Jack's trust (details are provided in the Supplementary Material). Thus, $\widehat{t_{Jack}}$ is given by:

$$\widehat{t_{Jack}} = \begin{cases} (\sum_{i=1}^{k} o_i)/k & \text{if } \overrightarrow{o^j_j} \neq \oslash \\ (\prod_{i=1}^{k}(1 - p_{s,i}))/2 & \text{if } \overrightarrow{o^j_j} = \oslash \end{cases} \quad (6)$$

Considering one sample set j and solving the corresponding MLE problem, the BS obtains an estimate $\widetilde{t^j_{Jack}}$. We do not use this value directly. Instead we use an exponential weighted average over recent sample *sets* to compute a trust value for Jack. We use a *sliding window* approach to assign observations to sets. That is, the first sample set consists of the observations (packet transmissions) indexed by $\{1, 2, ..., k\}$, the second sample set consists of the observations $\{2, 3, ..., k+1\}$ and so on. By using sliding windows (rather than non-overlapping ones), TPF is rendered robust, in the sense that it does not need to wait for a large period of time before obtaining additional sample sets for the estimation of Jack's trust. Waiting for k more additional observations, could required plenty time (depending on the value of k, the density of the users as well as the traffic patterns in the network) and could render older sample sets *stale*. After obtaining the estimation from the j^{th} sample set the assessed value for Jack's reporting trust is updated using the following equation:

$$t^j_{Jack} = \beta \cdot t^{j-1}_{Jack} + (1 - \beta) \cdot \widetilde{t^j_{Jack}} \quad (7)$$

Note here that our model (Equation (1)) incorporates only one parameter. This is enough to cover all cases

7. Even if this assumption does not hold, MLE can still provide us with the average trust on Jack's UE during the k transactions of the sample set.

of misreports (Table 3). For instance, when Jack is acting selfishly, even if the NACK reported is correct, the packet reception would not have failed if he had reported the actual CQI. Thus, our framework will impose a reduction in $t_{i,Jack}$. Similarly, when Jack aims to cause DoS, reporting many NACKs, (regardless of the correctness of the CQI report) will cause a degradation in $t_{i,Jack}$ due to the "failed" observations.

4.2 Trustworthy Proportional Fair scheduler

We now examine how to use the trust estimation to improve the PF scheduler. In particular, we scale the scheduling priority value for Jack's UE using t^j_{Jack}:

$$p^*_{Jack}(t) = t^{j(t)}_{Jack} \cdot \frac{CQI_{Jack}(t)}{A_{Jack}(t-1)} \quad (8)$$

where the superscript $j(t)$ denotes that the estimation of the trust on Jack's UE as per Equation (7) is done using all complete sample sets until time t. Observe that in effect the scheduler reduces all priority values (since for all users u, $t^{j(t)}_u \leq 1$), but the reduction is greater for untrustworthy users (with small $t^{j(t)}_u$). Only completely trustworthy users experiencing *perfect* channel conditions, i.e., those with $t^{j(t)}_u = 1$, will not observe a reduction in priority.

As it might be clear from the above discussion the trust value of a well behaved user who experiences packet losses due to wireless induced effects can degrade. Nevertheless, our evaluations indicate that this reduction is not critical, since every user is susceptible to similar losses. Thus, under benign conditions the trust values of all users are very similar, retaining the relative priorities of users unchanged. Consequently, there is no performance degradation in benign settings.

5 Evaluations

In order to quantify the effects of misbehaviors and evaluate our proposed scheme, we implement a discrete event simulator in MATLAB. The design of the simulator and the choice of the various physical layer parameters are based on the studies presented in [8], [14] and [15], as well as on the 3GPP standard (e.g., [16] and [9]).

5.1 Simulation Environment

Since we are interested in examining the robustness of the scheduling mechanism, the simulator does not implement higher layer functionalities (e.g., transport layer). We consider a single cell and for each UE we sample its distance from the BS from a uniform distribution U (see Supplementary Material). Using the lognormal shadow fading propagation model [17] [18] — briefly described in the following — we estimate the received power at the UE from the BS. Assuming a constant intra- and inter- cell interference, we can estimate the SINR and consequently, we compute the CQI. We further develop a measurement-driven model, to capture and incorporate in our simulator the temporal variations of the CQI values. We use the CQI-TBS mapping of Category 7/8 of UE to decide on the transmission block size [9]. Note here that, for the purpose of our study the actual block sizes used (i.e., the actual category and consequently the transmission rates) are not of great importance since we are mainly interested in the medium access opportunities of each user.

Propagation Model: In order to calculate the received power P_r at distance r with transmission power P we use the lognormal shadow fading model. In particular, the model computes P_r as follows:

$$P_r = \frac{P}{r^\alpha} \cdot Y, \quad (9)$$

where α is the path loss exponent and Y is a random variable that is log-normally distributed. The random variable Y models the shadow fading effects and it has a mean value of one and a standard deviation equal to the shadow fading variation. The above model has been shown to be reasonably accurate [17], [18]. The details of our measurement-driven temporal model are provided in the Supplementary Material.

5.2 Simulation Results

In Section 3 we have quantitatively described the misreporting behaviors and their effects. In the following we quantify the extent to which the normal network operations are disrupted from such behaviors, and the network gains possible from our solution. Furthermore, it is crucial to make sure that there are no "side effects" accompanying TPF, that is, unwanted degradation under benign settings. Finally, we are interested into examining the accuracy of our trust assessment scheme as a stand-alone module, since this inference engine can be possibly integrated with other network functionalities in the future that require a trust estimation.

The effect of misbehaviors on PF scheduler: Our first set of evaluations aims at quantifying the effect of the different types of misreporting behaviors on the PF scheduler which incorporates no trust features. We perform simulations with 100 users in total, varying the number of misbehaving nodes. We simulate 20000 timeslots. We examine the performance of PF scheduler under different types of behaviors. In particular, we examine the various combinations of correct/fake CQI/ARQ reports. For the ARQ reports, a misbehaving UE (say Jack's) with pre-defined trust x has the following choices: (i) always transmit a NACK, (ii) always transmit the correct feedback (e.g., selfish behavior), or (iii) transmit a NACK with probability x (without considering the actual outcome of the downlink transmission). For the CQI reports, Jack either reports the real link quality or decides on a fake CQI based on the operations presented in Section 3.1.

Figure 1 presents the percentage of timeslots occupied by the set of misbehaving nodes. As one can observe under benign settings, the set of "misbehaving" nodes

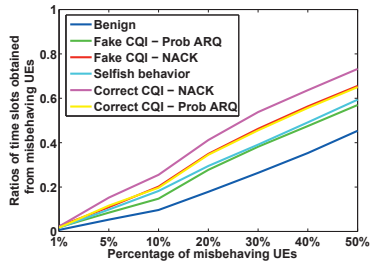


Fig. 1. *Misreporting* UEs can obtain a large - unfair - number of timeslots.

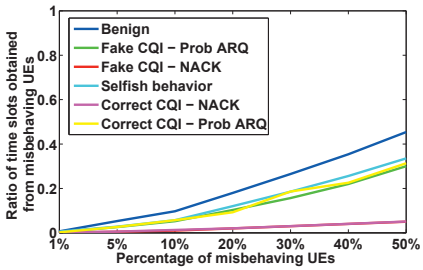


Fig. 2. TPF contains the effects of misbehaving UEs.

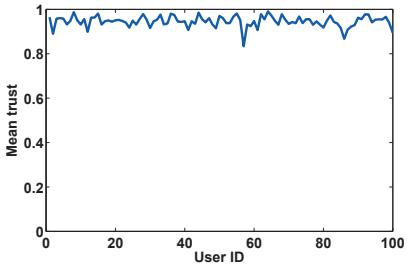
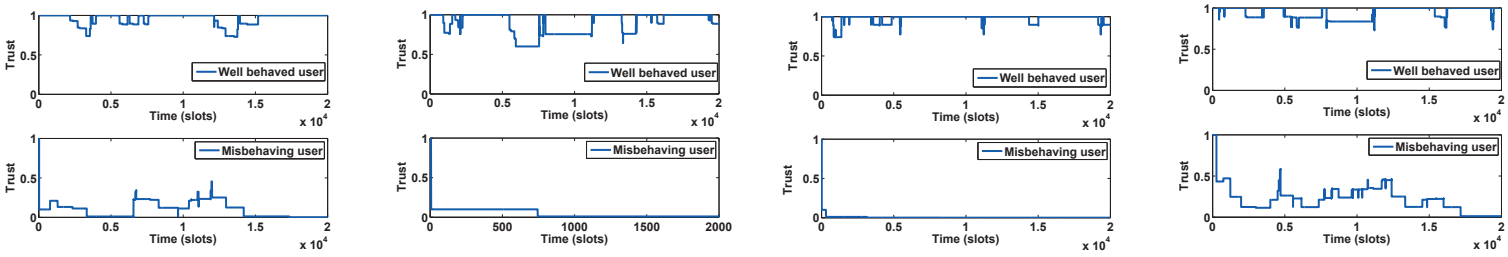


Fig. 3. Under benign settings the packet losses due to wireless effects affect the mean trust of all UEs to the same degree.



(a) Fake CQI - Prob ARQ (b) Fake CQI - NACK (c) Correct CQI - NACK (d) Correct CQI - Prob ARQ

Fig. 4. Misbehaving users exhibit significantly lower trust as compared to well behaved ones.

obtain their fair share of the medium as expected. However, when these nodes deviate from the expected behavior they can obtain a significant portion of the time slots, causing starvation to the rest of the users. Surprisingly, the most devastating attacks include reporting the correct CQI information. When reporting the actual link quality, the update in the throughput A_i is lower as compared to the case when the reported CQI is high. Thus, the priority values of the malicious nodes do not significantly decrease. In other words, the increase in the current priority value of a malicious user is smaller as compared to its consequent reduction due to the throughput update. Reporting fake CQI and always NACK leads to high degradation as well, since every time a malicious user is scheduled, he occupies L_{max} timeslots (L_{max}=3 in our case. For details see the Supplementary Materials). The extent of the effects is larger as the number of misbehaving nodes increase, as one might have expected.

In this work, we have mainly focused on malicious UEs that perform DoS attacks. It is also interesting to consider selfish users, whose goal is to increase their throughput as mentioned before. A selfish UE reports fake CQI (to increase his priority) but correctly reports the ARQ feedback, since it is not interested in retransmission of correctly received packets. As we observe from Figure 1 such UEs are able to obtain an unfair share of the time slots as well. However, a key question is: "Are these slots *successful* in delivering data?" Table 4 shows the fraction of the obtained timeslots from the selfish user(s) that were used for retransmissions. Essentially, this number is representative of the packet loss rate over the time slots obtained from the selfish users. Even though the selfish users are able to be scheduled in more timeslots than what is dictated from their fair share, these timeslots are wasted since no new or actual data are received.

Fraction of selfish users	Fraction of obtained slots used for retransmissions
0.01	0.8246
0.05	0.9485
0.1	0.9446
0.2	0.8867
0.3	0.8996
0.4	0.8676
0.5	0.8783

TABLE 4
Selfish users waste many time slots for retransmissions

The above results should have been expected from our discussion in Section 3.1. In particular, the reported link quality is much higher as compared to the actual one. Hence, to reiterate, the BS transmits a larger TBS than the one that can be sustained at the downlink, resulting in packets in error. Therefore, **reporting higher CQI values is not beneficial for the selfish users**; while it increases their service time, it significantly degrades their

throughput due to high rate transmissions.

Before examining the performance of TPF, we would like to connect the above results with our qualitative description in Table 3. In particular, we see that when the malicious UE adopts a "Correct CQI - Fake ARQ" policy, its effect is the severest, regardless of whether the fake ARQ is probabilistic or not. Furthermore, the case of both fake CQI and fake ARQ attack has also severe effects on the network, while the most *mild* threat, is the selfish behavior ("Fake CQI - Correct ARQ" report). Nevertheless, note here that even with the latter, the selfish UE can still obtain a very large fraction of time slots, leading legitimate users to starvation.

TPF's performance: Regardless the objectives of the misbehaving nodes (i.e., DoS or higher throughput) providing fake reports leads to the starvation of the well behaved UEs as seen from the above results. In the following we want to examine the performance of TPF both under benign settings as well as under the presence of misbehaving users. Using the same simulation setup as above we obtain Figure 2.

Let us first examine how TPF fares under benign operations. As alluded to above, packet losses due to wireless induced effects can reduce our trust on the UE. In spite of that, we observe that this does not affect the performance of the scheduler under benign settings. In particular, the nodes obtain their fair share in terms of timeslots. Since every UE observes packet losses with similar probabilities (recall that the BS's select transmission rates and packet sizes with the goal of achieving a successful packet reception with probability 0.9), all of them observe similar degradation of trust. Figure 3 presents the average trust of each user under benign settings. As we can observe the packet losses due to wireless effects reduce the trust of the UEs below 1, however they all exhibit similar values. Hence, the priority values p are still dictated by the link quality and the average throughput.

With regards to misbehaviors, TPF significantly contains their impact in all cases. Malicious UEs cannot obtain their fair share due to the significantly lower trust values as compared with that of the legitimate users. This translates to much lower priority values even with fake CQI reports. Figure 4 depicts representative trust time traces for a well behaved and a misbehaving node for the different type of malicious strategies.

As we can see the trust values of the malicious UE are significantly lower over the simulation period. While the initial state of the system is to "trust everyone", the trust value of the misbehaving users drops fast. Especially for the case of an aggressive node with respect to the ARQ feedback (i.e., always reporting a NACK), its trust values are close to 0 for the whole period. A less aggressive attack (i.e., probabilistically reporting a NACK) results into oscillations of the assessed trust, since users obtain credit for the ACKs reported. However, their overall behavior leads to low priority values, resulting in the mitigation of the attack.

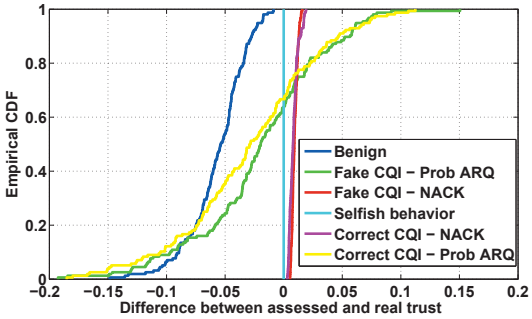


Fig. 5. Our trust assessment engine exhibits high accuracy.

Accuracy of our trust inference scheme: Next we want to examine the accuracy of our trust assessment mechanism. Every misbehaving node has an *a priori* trust value associated with him (*actual* trust). Figure 5 presents the Cumulative Distribution Function (CDF) of the difference between the mean value of the assessed trust $\bar{t}$ of a user and his actual trust t^*, $d = \bar{t} - t^*$.

For the majority of the cases ($> 95\%$), the absolute value of d is less than 0.1. Especially for the cases of aggressive misbehaving nodes — reporting always a NACK — the inference engine can estimate the trust value with extremely high accuracy (e.g., $d \approx 0$ with high probability). The observations serving the input to the MLE framework are consistent in this case (always 0) and this drives the inferred trust to low values. For the cases of more mild misbehaviors — probabilistic ARQ feedback — the occasional positive observations (i.e., ACK reports) *confuse* the estimation mechanism and thus the accuracy is slightly reduced. Nevertheless, note here that the accuracy is still high as it can be deduced from our results; indeed, the difference is less than 0.2 with probability more than 90%, and rarely more than 0.25.

6 Conclusions

In this work we study a broad class of reporting misbehaviors in cellular networks. The PF scheduler utilized by the network expects correct feedback from each user with respect to (i) the downlink quality and (ii) the success or failure of the downlink transmissions. Currently, the scheduler blindly trusts these reports from the UEs. We have shown that this opens a security backdoor for malicious users who can manipulate these reports. We analyze the impacts of various types of misbehaviors, both qualitatively and quantitatively and we propose a variation of the scheduler called TPF that integrates a trust module to mitigate the impacts. The trust module utilizes well established statistical frameworks for inferring the *reporting trust* of each user and scaling his priority values for scheduling downlink transmissions accordingly. Our evaluations indicate that the accuracy of our assessment scheme is high; the absolute difference between the estimated and the real trust value is

smaller than 0.1 for 95% of the cases examined. TPF is further shown to be capable of completely mitigating the adversarial effects, while not degrading the performance under benign settings.

REFERENCES

[1] O. Whitehouse and G. Murphy. Attacks and Counter Measures in 2.5G and 3G Cellular IP Networks. In *Stake Research Report*, March 2004.

[2] R. Racic, D. Ma, H. Chen, and X. Liu. Exploiting Opportunistic Scheduling in Cellular Data Networks. In *NDSS*, 2008.

[3] Udi Ben-Porat, A. Bremler-Barr, and H. Levy ad B. Plattner. On the Vulnerability of the Proportional Fairness Scheduler to Retransmission Attacks. In *IEEE INFOCOM*, 2011.

[4] K. Pelechrinis, P. Krishnamurthy, and C. Gkantsidis. Towards a trustworthy pf scheduler for cellular data networks. In *IEEE GLOBECOM*, 2012.

[5] J. Bergman, M. Ericson, D. Gerstenberger, B. Goransson, J. Peisa, and S. Wager. HSPA Evolution - Boosting the performance of mobile broadband access. In *Ericsson Review, No 1*, 2008.

[6] T. Bu, L. Li, and R. Ramjee. Generalized Proportional Fair Scheduling in Third Generation Wireless Data Networks. In *IEEE INFOCOM*, April 2006.

[7] H. Zheng and H. Viswanathan. Optimizing the ARQ Performance in Downlink Packet Data Systems With Scheduling. In *IEEE Transactions on Wireless Communications*, March 2005.

[8] F. Brouwer, I. de Bruin, J.C. Silva, N. Souto, F. Cercas, and A. Correia. Usage of Link-Level Performance Indicators for HSDPA Network-Level Simulations in E-UMTS. In *IEEE Spread Spectrum Techniques and Applications*, 2004.

[9] 3GPP TS 25.306 v9.0.0. http://www.3gpp.org/ftp/Specs/html-info/25306.htm.

[10] D. Kim and Y-C. Hu. A Study of False Channel Condition Reporting Attacks in Wireless Networks. In *SecureComm*, 2010.

[11] S. Bali, S. Machiraju, H. Zang, and V. Frost. A Measurement Study of Scheduler-Based Attacks in 3G Wireless Networks. In *PAM*, April 2007.

[12] D. McKnight and N. Chervany. The Meanings of Trust. In *Management Information Systems Research Center, University of Minnesota, Working Paper 96-04*, 1996.

[13] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice Hall, ISBN 0-13-345711-7.

[14] L. Klockar, A. Simonsson, F. Gunnarsson, and A. Borg. Channel Characterization and HSDPA Bit Rate Prediction of a Dense City Network. In *in IEEE VTC*, 2009.

[15] M. Folke, S. Landstrom, U. Bodin, and S. Wanstedt. Scheduling Support for Mixed VoIP and Web Traffic over HSDPA. In *in IEEE VTC*, 2007.

[16] 3 GPP TS 25.214 V5.5.0. Physical layer procedures (FDD), Release 5, 2005.

[17] S. Zvanovec, P. Pechac, and M. Klepal. Wireless LAN Networks Design: Site Syrvey or Propagation Models? In *Radioengineering, Vol. 12, No. 4*, Dec. 2003.

[18] T. S. Rappaport. *Wireless communications principles and practices,*. Prentice Hall, 2002.

Konstantinos Pelechrinis Konstantinos Pelechrinis received his PhD from the Computer Science department of University of California, Riverside, in 2010. Previously he obtained his MSc degree from the Computer Science department of University of California, Riverside in 2008 and the diploma of Electrical and Computer Engineering from the National Technical University of Athens, Greece, in 2006. He is an Assistant Professor at the SIS faculty of the University of Pittsburgh since Fall 2010. He has also held research positions at LANL, Thomson Research Labs Paris and MSR Cambridge. He was a visiting researcher at the University of Thessaly during Fall 2008. His research interests include wireless networking, especially security - related issues that span the full protocol stack. He is involved in protocol design, real world experimentation and performance analysis. He is also interested in mathematical foundations of communication networks.

Prashant Krishnamurthy Prashant Krishnamurthy is an associate professor in the School of Information Sciences, University of Pittsburgh, Pennsylvania, where he regularly teaches courses on wireless networks and cryptography. From 2000 to 2005, he also served as the chair of the IEEE Communications Society Pittsburgh Chapter. His research interests include wireless network security, wireless data networks, position location in indoor wireless networks, and radio channel modeling for indoor wireless networks. He is a member of the IEEE.

Christos Gkantsidis Christos is a researcher in the Systems and Networking Group in Microsoft Research, Cambridge, UK. He holds a Ph.D. from Georgia Institute of Technology, Atlanta, GA, USA, and a bachelor from University of Patras, Greece, both in computer science. He is interested in big data analytics, analysis and modelling of communication networks, content distribution networks, and wireless networking.