Exploiting and Defending Opportunistic Scheduling in Cellular Data Networks

Radmilo Racic, Denys Ma, Hao Chen, *Member, IEEE*, and Xin Liu, *Member, IEEE*

Abstract—Third Generation (3G) cellular networks take advantage of time-varying and location-dependent channel conditions of mobile users to provide broadband services. Under fairness and QoS constraints, they use opportunistic scheduling to efficiently utilize the available spectrum. Opportunistic scheduling algorithms rely on the collaboration among all mobile users to achieve their design objectives. However, we demonstrate that rogue cellular devices can exploit vulnerabilities in popular opportunistic scheduling algorithms, such as Proportional Fair (PF) and Temporal Fair (TF), to usurp the majority of time slots in 3G networks. Our simulations show that under realistic conditions, only five rogue device per 50-user cell can capture up to 95 percent of the time slots, and can cause 2-second end-to-end interpacket transmission delay on VoIP applications for every user in the same cell, rendering VoIP applications useless. To defend against this attack, we propose strengthening the PF and TF schedulers and a robust handoff scheme.

Index Terms—Security, opportunistic scheduling, proportional fair, temporal fair, handoff.

1 Introduction

3G cellular networks, such as High Speed Downlink Packet Access (HSDPA) [1] and Evolution-Data Optimized (EV-DO) [2], provide broadband-like downlink speed to enable applications, such as Voice-over-IP (VoIP). The specification for 3G cellular data services recommends implementing an *opportunistic scheduler*. An opportunistic scheduler uses multiuser diversity—the fading and shadowing of cellular users within a single cell—to optimize bandwidth efficiency. Both HSDPA and EV-DO use an opportunistic scheduler in the downlink to profit from multiuser diversity. To achieve this goal, many networks require mobile devices to participate in managing network services. However, since mobile devices are outside the control of the network administrators, networks should not trust them to manage network operations [3]. Unfortunately, this principle is often violated, as in the case of the popular opportunistic scheduling algorithms, *Proportional Fair* (PF) [1], [2], [4], [5], [6], [7] and *Temporal Fair* (TF) [8], [9].

Apropos, we discovered two vulnerabilities:

1. PF and TF schedulers trust channel condition reports from mobile devices without verification.
2. Both schedulers guarantee fairness only within a single cell.

A malicious mobile device can exploit the first vulnerability by reporting bogus channel conditions, and can exploit the second vulnerability by initiating unnecessary handoffs to circumvent the per-cell fairness guarantee. As a result, the attack can usurp a large number of time slots at the expense of the other users in the same cell. Our simulation shows that only one attacker per 50-user cell can occupy up to 92 percent of available time slots persistently, depending on the scheduling algorithm used. To put it in another perspective, when users are running VoIP applications, one attacker per cell can perpetuate a 1-second end-to-end interpacket transmission delay for every other user, while five attackers per cell can perpetuate a 2-second delay. Since any delay longer than 0.4 second would disrupt VoIP [10], this attack would render VoIP useless.

In addition to describing and analyzing the attack, we discuss two defense strategies. First, we propose to augment the PF and TF schedulers with priority queue and round-robin to mitigate the attack. However, as the current PF and TF schedulers operate within a single cell, they cannot guarantee long-term fairness to mobile devices that can hand off freely across cells, resulting in the second vulnerability mentioned above. Therefore, we propose a robust handoff procedure that ensures graceful handoff for honest users while preventing attackers from usurping bandwidth. Our simulation shows that under this handoff procedure, the percent of time slots that attackers can obtain, no matter how many arbitrary handoffs they initiate, is close to what they can get in a single cell. This result demonstrates that our robust handoff procedure effectively prevents attackers from gaining advantage by initiating arbitrary handoffs.

We make the following contributions:

- We identify vulnerabilities in two popular opportunistic schedulers. We analyze a series of attacks mathematically as well as through simulations to demonstrate that they could devastate victim mobile users by causing persistent delays, lowering throughput, and disrupting certain applications.
- We propose defense strategies against these attacks. Our simulation shows that our proposed robust handoff procedure effectively removes attackers' advantage during their malicious handoffs.

2 3G Data Networks

One of the principal objectives of designing mobile devices is to improve functionality while reducing network-wide

- *The authors are with the Department of Computer Science, University of California, Davis, Davis, CA 95616.*
  E-mail: {rracic, denys.ma}@gmail.com, {hchen, liu}@cs.ucdavis.edu.

Manuscript received 9 Oct. 2008; revised 8 June 2009; accepted 4 Aug. 2009; published online 12 Aug. 2009.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2008-10-0401.
Digital Object Identifier no. 10.1109/TMC.2009.146.

1536-1233/10/$26.00 © 2010 IEEE Published by the IEEE CS, CASS, ComSoc, IES, & SPS

component count, complexity, and cost [11]. In this process, however, 3G cellular networks grant unwarranted trust to mobile devices, allowing them to report arbitrary channel conditions and to initiate handoffs at their discretion. By exploiting these vulnerabilities, malicious mobile devices can disrupt other mobile users severely.

2.1 HSDPA

Cellular providers have developed two new data services, HSDPA and EV-DO, to meet the increasing demands for mobile technologies as alternatives to traditional wired communications. In both services, the downlink utilizes time division multiplexing (TDM) by dividing the channel in time slots, or Transmission Time Intervals (TTIs). The *scheduler* at each base station selects a single user to transmit at each TTI. Both services rely on two main techniques to increase efficiency in the downlink direction: *link adaptation* and *fast retransmissions*. Link adaptation is a data rate regulating mechanism in which mobile devices report to base stations their quasi-instantaneous downlink channel quality information, *channel quality indicator* (CQI). Base stations can then establish data rate contingent on channel conditions: the better the channel condition, the higher the data rate [12]. Fast retransmissions (part of the Hybrid Automatic Repeat Request (HARQ) manager) allow mobile devices to NACK each erroneous downlink packet (and request a retransmission) from its base station instead of the sending server.

2.2 Opportunistic Scheduling

Channel conditions of cellular mobile devices are time-varying and location-dependent due to fading and shadowing. This causes the multiuser diversity effect [13]: since many users fade independently, at any given time, some subset of users will likely have strong channel conditions. As we have already stated, better channel conditions imply higher data rates. On the one hand, a good scheduling scheme can recognize and exploit favorable channel conditions of certain users to achieve higher utilization of wireless resources. On the other hand, the potential to exploit favorable channel conditions of a subset of users introduces a trade-off problem between resource efficiency and fairness. A very popular opportunistic scheduler is PF [6], [7], whose goal is to maximize the product of the throughput delivered to all users [14], [15]. In fact, Kushner and Whiting [16] have shown that PF is not an ad hoc algorithm, but actually corresponds to a maximization problem. Another interesting opportunistic scheduler is TF [8], whose goal is to maximize the average system performance, given the time fraction assignment. Both PF and TF attempt to strike a balance between throughput and fairness within a single cell [4], [8], [14], [15], [17], [18].

Channel quality indicators. Since instantaneous channel conditions derive the instantaneous data rates of mobile devices [19], mobile devices constantly measure and report their CQIs to their base stations. In particular, at every TTI, an opportunistic scheduler at a base station selects a user (or a subset of users) with a relatively good channel condition to transmit while maintaining predefined QoS or fairness constraints. By scheduling the users with the best channel condition, opportunistic schedulers utilize the shared channel efficiently and often achieve higher network performance than other schedulers, such as round-robin.

In the current HSDPA specification, each mobile device periodically measures its instantaneous channel conditions through pilot signals,[1] estimates the achievable data rate under its channel condition (denoted as $CQI_i(t)$ for user i at time t), and sends the information back to the base station. The CQI value is calculated by an iterative algorithm that takes as input the downlink channel quality and a number of tunable parameters. The algorithm iterates with varying parameter combinations until the block error rate is less than 10 percent. Note that it is up to the mobile device to upload these reports to the base station at its own timing. According to the specification [20], [21], the CQI report cycle can happen every 1, 2, 4, 5, 10, 20, 40, or 80 TTIs.

2.2.1 Proportional Fair

PF is a compromise scheduling algorithm. It tries to strike a balance between achieving maximum network throughput and ensuring fairness. In doing so, PF scheduler maximizes the product of throughputs delivered to all users [22]. PF selects a user i to schedule at time slot t based on the following criterion:

$$i = \arg\max_{1\le k\le N} \frac{DRC_k(t)}{R_k(t)} = \arg\max_{1\le k\le N} \frac{min\{CQI_k[t], \frac{B_k[t]}{TTI}\}}{R_k(t)}, \quad (1)$$

where $DRC_k(t)$ is the currently supported data rate, $B_k[t]$ is the buffer size, and TTI is the Transmission Time Interval. For simplicity, we assume that all users always have outstanding data at the base station, therefore eliminating the term $B_k[t]/TTI$. $R_i(t)$ is the average throughput of user i up to time t. The base station estimates $R_i(t)$ as follows:

$$R_i(t) = \begin{cases} \alpha CQI_i(t) + (1-\alpha)R_i(t-1), & \text{if } i \text{ is scheduled,} \\ (1-\alpha)R_i(t-1), & \text{otherwise,} \end{cases} \quad (2)$$

where α is a network parameter describing the weight of the current time slot toward the average. Strictly speaking, $\alpha = 1/t_c$, where t_c is the time window within which the average rate $R_i(t)$ is calculated. A typical t_c is 2-second (1,000 slots).

While current 3G standards do not specify a particular opportunistic scheduler, PF is the most popular both in the research community [23], [24], [25], [26], [27], [28], [29] and industry [1], [2], [4], [5], [6], [7], [30]. Networks may implement a modified PF. For instance, a PF scheduler may apply code multiplexing by scheduling multiple users within the same TTI. Researchers have also proposed combining the PF scheduler with a priority queue or the round-robin scheduler. For the rest of the paper, however, we will focus on the original PF, discussed in detail above.

2.2.2 Temporal Fair

TF algorithm provides another way of balancing system and individual user performance. Its goal is to maximize the average system performance by exploiting time-varying channel conditions, given the time fraction requirements of

[1] A continuous signal, sent by the base station over pilot channels, to facilitate device synchronization and signal strength measurement.

all users [8]. Let r_i denote the predetermined minimum fraction of time when the user i should transmit. $r_i \geq 0$, $\sum_{1=1}^{N} r_i \leq 1$, where N is the number of users in the cell. The network determines each r_i by the user's class, the user's pay level, and the current channel conditions. Additionally, let $U(\vec{t}) = (U_1(t), \ldots, U_N(t))$ be the performance vector at time slot t, where $U_i(t)$ is user i's performance if he transmits at time t. $U_i(t)$ can be any predetermined function of the channel condition $CQI_i(t)$. For the rest of the paper, we assume that $U_i(t) = CQI_i(t)$. The scheduling problem is, given the performance vector $\vec{U}$, to determine the scheduling policy $Q : \vec{U} \rightarrow \{1, \ldots, N\}$, i.e., which user should be scheduled in the next time slot. Q is opportunistic since it can use the performance vector to decide which user to schedule. TF scheduler's goal is to maximize the average system performance under individual users' resource sharing requirement:

$$\max E\big(U_{Q(\vec{U})}\big) \text{ s.t. } P(Q(U) = i) \geq r_i, \forall i. \tag{3}$$

An optimal Q is defined as

$$Q(\vec{U}) = \arg\max_i (U_i + v_i), \tag{4}$$

where v_is are the offset parameters to satisfy the fairness constraint:

1. $\min_i(v_i) = 0$.
2. $\forall i \in [1..N], P\{Q(\vec{U}) = i\} \geq r_i$.
3. $\forall i \in [1..N], P\{Q(\vec{U}) = i\} > r_i$, then $v_i = 0$.

Informally, to satisfy the fairness requirements, policy Q schedules the *relatively best* user to transmit. A user i is relatively best if $U_i + v_i \geq U_j + v_j, \forall j \in [1..N]$. Put another way, if a user i experiences chronic unfavorable channel conditions—e.g., it is far away from the base station—the user has to take advantage of some other users (whose $v_j = 0$) to satisfy its fairness requirement, i.e., $v_i > 0$. To maximize the overall system performance, TF gives such users only as much help as needed to reach their predetermined share of time slots. If a user satisfies $P\{Q(\vec{U}) = i\} > r_i$, the user i has already been allotted its minimum share, so it cannot take advantage of other users, i.e., $v_i = 0$.

Opportunistic scheduler gain. Opportunistic scheduling gain $G(N)$ illustrates the performance gain of an opportunistic scheduling scheme over that of the nonopportunistic one, namely, round-robin. Typically, the larger the number of users sharing the same channel, the larger the gain. For example, when users experience Rayleigh fading with statistically identical and independent relative channel conditions, $G(N) \approx \log(N)$.

2.3 Handoffs

Cellular networks utilize *handoffs* to transfer connections from one base station to another. A mobile device continuously monitors candidate base stations with stronger signal strength using pilot signals. The base station controller, upon receiving pilot measurement reports, determines if the mobile device will benefit from a handoff. If so, the base station controller initiates a handoff procedure by instructing the mobile device to hand off to another base station [5].[2] There are two types of handoffs: soft and hard handoffs. In a hard handoff, the network drops the connection to the current base station before initiating a new one. In a soft handoff, a mobile device can have connections from several base stations simultaneously. Our attacks apply to soft as well as hard handoffs.

3 Overview of Attacks

Opportunistic schedulers for 3G networks require mobile device to participate in network management functions. However, attackers can modify mobile devices to perform actions that are undesirable to the providers, even when providers attempt tamper-proof techniques [6], [31], [32], [33]. For instance, attackers can modify their laptops' 3G PC cards, either through the accompanying SDKs [34] or the device firmware [35]. By trusting all mobile devices for network management, a system that implements either PF or TF scheduler suffers from two vulnerabilities, discussed in the following.

3.1 Fabricated CQIs

Opportunistic schedulers base their scheduling decisions on CQIs reported by mobile devices without verification. By reporting fabricated CQIs, malicious mobile devices can manipulate the scheduler in their own favor. Let us consider a naïve attack on PF and TF, respectively, with one attacker. In the PF variety of the attack, the malicious mobile device reports an inflated CQI such that its ratio of currently supported data rate to average data rate is the highest among all the devices in its cell; therefore, ensuring that it will be scheduled in the next time slot. To obtain consecutive time slots, the attacker must report monotonically increasing CQIs (because its average throughput is increasing, while other users' throughput is decreasing, according to (2)) until its reported CQI exceeds the range of CQI values. In the TF variety of the attack, a malicious mobile device starts with an inflated CQI. Then, it continues misrepresenting its channel conditions and reporting monotonically increasing CQIs. This action causes the scheduler to keep decreasing the malicious device's offset as well as its allotted time share to satisfy the overall fairness.

3.1.1 *Mathematical Analysis of the PF Attack*

It is difficult to calculate the precise number of consecutive time slots that the attacker can get, because the number depends on the channel conditions of all the users in the cell. However, we can estimate an upper bound of this number by considering a simplified situation where each user has the same CQI. We assume that each user always has outstanding data at the base station. First, we calculate the average throughput of a user. Let $R_i(t)$ be the average throughput of user i at time slot t. Recall from Section 2.2, $R_i(t)$ is determined by whether the user is scheduled (2). Since we assume that each user has the same CQI, the PF scheduler becomes a round-robin scheduler, where each user is scheduled once every N slots (N is the number of users in the cell). For example, if user i is scheduled at time slot s, he will not be scheduled until time slot $s + N$. Therefore, user i's average rate $R_i(t)$ maximizes at time slot s, and minimizes at the time slot $s + N - 1$. According to (2),

$$R_i(s) = (1 - \alpha)^N R_i(s - N) + \alpha CQI. \tag{5}$$

[2] Note that EV-DO implements mobile initiated handoffs instead.

Let us consider a steady state, where $R_i(t) = R_i(t + kN)$ for all integer k. In this case, $R_i(s) = R_i(s - N)$. Using this equality in (5), we have

$$R_i(s) = \frac{\alpha CQI}{1 - (1-\alpha)^N} \approx \frac{CQI}{N}, \tag{6}$$

where $R_i(s)$ is the user i's maximum throughput. His minimum throughput is

$$\begin{aligned} R_i(s-1) = R_i(s+N-1) &= (1-\alpha)^{N-1} R_i(s) \\ &\approx (1-\alpha)^{N-1} \frac{CQI}{N}. \end{aligned} \tag{7}$$

Let $C(t) = \max_i\{CQI/R_i(t)\}$ be the maximum CQI-to-throughput ratio at time t among all the users. In the steady state, $C(t)$ becomes a constant C, which is

$$C = \frac{CQI}{R_i(s-1)} \approx \frac{N}{(1-\alpha)^{N-1}}. \tag{8}$$

Next, we describe a strategy for the attacker to obtain consecutive time slots. To obtain time slot 1, the attacker i must report a $CQI_i(1)$ such that $CQI_i(1)/R_i(0) \geq C(0)$. After time slot 1, $C(1) = C(0)/(1-\alpha)$, because for each victim user j, its CQI remains constant, but its average throughput R_j has been scaled down by a factor of $(1-\alpha)$. Therefore, to obtain time slot 2, the attacker i must report $CQI_i(2)$ such that $CQI_i(2)/R_i(1) \geq C(1) = C(0)/(1-\alpha)$. Subsequently, at time t, the attacker must claim $CQI_i(t)$ such that $CQI_i(t)/R_i(t-1) \geq C(0)/(1-\alpha)^{t-1}$. The attacker can obtain consecutive time slots until the required $CQI_i(t)$ exceeds CQI_{max}, the maximum value of CQI. Therefore, the maximum number of consecutive time slots that the attacker can obtain is the maximum integer t_0 that satisfies

$$CQI_{max} \geq \frac{C}{(1-\alpha)^{t_0-1}} R_i(0) \cdot \Pi, \tag{9}$$

where Π is

$$\Pi = \prod_{k=1}^{t_0-1} \left(\frac{\alpha C}{(1-\alpha)^{k-1}} + (1-\alpha) \right).$$

Equation (9) shows that the maximum number of consecutive slots an attacker can obtain (t_0) depends on the attacker's initial average throughput ($R_i(0)$), maximum CQI (CQI_{max}), the PF parameter α, and the number of users in the cell (N). Since CQI_{max} and α are set by the system, they are out of the attacker's control. The attacker also cannot easily control N, the current number of users in the cell. However, the attacker does have control over $R_i(0)$, its average throughput at the beginning of the attack. Equation (9) shows that the smaller the value $R_i(0)$, the larger the value t_0. Finally, this model is simplified, assuming that all victim users have the same, consistent CQI. When users have time-varying channel conditions, (9) provides an upper bound for estimating t_0.

3.2 Greedy Handoffs

Opportunistic schedulers are oblivious to mobile device handoffs. For example, when a mobile device hands off, the new base station does not retrieve the device's average data rate from its previous base station, but rather assigns an often small or average value as the device's initial average rate [28], [29]. To sustain the PF or TF attack, an attacker must report monotonically increasing CQIs, eventually causing the attack to stop when its reported CQI exceeds the maximum allowable CQI. However, if the attacker sits in the overlapping coverage area of multiple base stations, he may hand off to another cell to acquire a fresh, lower average data rate to continue the attack. Moreover, multiple malicious devices may cooperate to attack multiple cells simultaneously (Section 4.2.2). Note that by manipulating its CQI reports, a malicious mobile device can cause its base station to initiate a handoff.

4 Attack Analysis

4.1 Threat Model

Our threat model assumes the following:

1. Attackers control one or a few mobile devices that a cellular network has admitted and authenticated.
2. Attackers have modified their 3G mobile devices or PC cards such that they may report any CQI value to the base station and to trigger a handoff at any time.
3. Attackers can be physically located anywhere within cells under attack.

We believe that this threat model is realistic. Attackers can buy network-approved mobile devices (or PC cards with accompanying SDKs) and prepaid data plans directly from providers, or can spread worms to take over existing mobile devices. Prepaid data plans, in particular, minimize the risk of discovery and punishment. Previous research has demonstrated ways to modify mobile devices to perform different actions than intended by the providers, even when providers attempted tamper-proof techniques [31], [32], [33]. Note, however, that our threat model does not assume hacking into the network. Instead, our attack exploits vulnerabilities in the network's scheduler by *manipulating the information that malicious mobile devices report to the network.*

In the following sections, we describe the PF variety of attacks in detail. First, we describe a naïve attack, the intracell attack. Then, using this attack as a waypoint, we describe a more sophisticated and powerful attack, the intercell attack. Finally, to evaluate the intercell attack in a more realistic environment, we relax the requirement for the attackers to know the victim users' channel conditions. For TF scheduler, we can design a similar attack strategy, which we will describe at the end of this section.

4.2 Proportional Fair Attacks

4.2.1 Intracell Attacks

Consider a scenario where all attackers stay in the same cell. We assume that no user leaves or joins the cell during the attack. Although this assumption is not crucial to our attack, it simplifies our analysis. Additionally, for simplicity, we assume that the attackers know the channel conditions of all the users in the cell. Section 4.2.3 will describe an attack strategy which eliminates this assumption.

As we have stated in the previous section, a single attacker can obtain consecutive time slots until his reported CQI exceeds the maximum CQI value. Naturally, attackers can increase the number of consecutive time slots obtained by using multiple colluding attackers. We discuss three possible ways for the attackers to collude.

Sequential attack. Attacker with the smallest average throughput $R_i(t)$ starts the attack, while the other attackers lurk by reporting arbitrarily small CQIs to avoid being scheduled. When the active attacker's reported CQI exceeds the maximum CQI value, it stops the attack, while the attacker with the smallest average throughput takes over the attack.

Minimum CQI attack. Since the attack will stop when all attackers' reported CQIs exceed the maximum value, this scheme tries to slow the increment of the reported CQIs. At each time slot, each attacker, given its current average data rate, computes the CQI that it needs to obtain the next time slot. The attacker with the smallest computed CQI reports its CQI to the base station, while other attackers report arbitrarily low CQIs to continue lurking.

Delta CQI attack. This algorithm tries to slow the increment of calculated CQI values for upcoming slots. At each time slot t, each attacker i computes the increment $\delta_i(t)$ needed to its previous CQI. In other words, $\delta_i(t) = CQI_i(t) - CQI_i(t-1)$. The attacker with the smallest $\delta_i(t)$ then reports its CQI to the base station, while the other attackers report arbitrarily low CQIs to continue lurking.

Attack results. To verify the effectiveness of this attack, we ran simulations for 18,072 time slots, or about 36 seconds. This duration is long enough to evaluate the attack effects because we observed that all attacks stabilized well before the end of the duration. We repeated each simulation 100 times to average possible random effects. In simulating single-cell attacks, we chose parameters that are recommended by 3G and HSDPA specifications or that are commonly used by cellular networks. The PF scheduler has an $\alpha = 0.001$. Recall from Section 2.2 that α governs the maximum time a user can be starved. We assume 50 users in a cell. Each user quantized his channel condition into a CQI, an integer between 0 and 30, and reported the CQI to the base station. Each user's channel condition was a random variable following a Rayleigh distribution [36] with $\sigma = 3$ and an initial average rate of 0.5. In communications theory, Rayleigh distribution is widely used to model scattered signals that reach a receiver by multiple paths, e.g., in an urban environment [36].

A simulation with only one attacker showed that the attacker gained an average of 19 consecutive time slots, with a standard deviation of 2.77. Next, we simulated multiple attackers in the same cell. We varied the number of attackers from one to five and simulated each of the attack schemes in Section 4.2.1. Notice that the number of collective consecutive time slots obtained by the attackers increases almost linearly with the number of attackers. Among the three attack schemes, the Delta CQI scheme performed the best, where five attackers obtained 99 consecutive time slots.

Although 99 consecutive time slots (or 198 ms) occupied by the attackers will cause a delay to victim users, this delay is tolerable by many applications and protocols. Moreover, after the attack, the attackers must relinquish a large number (at least 2,000) of time slots to reset their average throughput low enough before they can attack again. Therefore, the intracell attack is unsustainable by itself. Fortunately (or unfortunately, depending on your position), we can exploit another vulnerability to devise a much more sustainable and effective attack, the intercell attack, to be discussed next in Section 4.2.2.

Even though the intracell attack is unsustainable by itself, it is an indispensable component of the intercell attack. In the intercell attack, when the attacker cannot acquire consecutive slots in a cell, it hands off to a neighboring cell. Since handoff requires at least one slot, the maximum percentage of slots (from the attacker's perspective) that the attacker acquires is $\frac{s}{s+1}$, where s is the number of consecutive slots that the attacker acquires in a single cell. To maximize the percentage, the attacker must maximize s, hence the goal of the single-cell attack.

4.2.2 Intercell Attacks

The PF scheduler ensures long-term fairness *within* a cell. By transgressing cell boundaries, attackers can gain unfair share of network bandwidth. Our single-cell simulations show that an attacker's reported CQI and average throughput increase very fast during an attack. When a large average throughput forces the attacker to report a CQI larger than the maximum value, the attack stalls. However, when a user joins a cell, the scheduler assigns a typically small value as the user's initial average throughput, since the network does not transfer users' average throughput across cells during handoff [29]. Therefore, when an attacker cannot acquire more slots because its average throughput is too high, it can induce a handoff to receive a smaller initial average throughput in the new cell. For example, consider two attackers M_A and M_B sitting in the overlapping area of cells C_A and C_B. Initially, M_A attacks C_A, and M_B attacks C_B. When one of the attackers fails to acquire consecutive slots, M_A hands off to C_B and M_B hands off to C_A to continue their attacks. Alternatively, consider a targeted cell attack. In this case, the attacker can use handoff as a bridge to reset its attack. In particular, he or she attacks the target cell as long as possible, handoffs to a neighboring cell and back immediately. This way, it resets the average throughput value and can continue the attack.

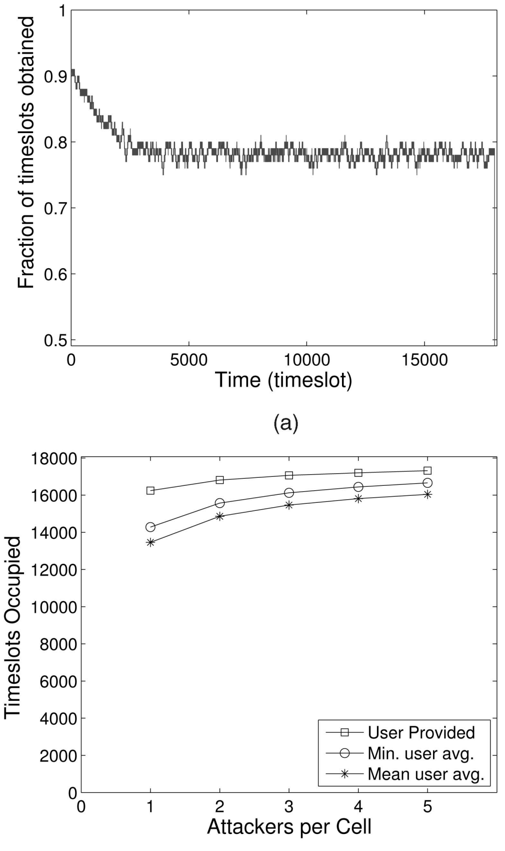
Since the choice of this initial value is unspecified, we explore three reasonable schemes that, although not all-inclusive, illustrate behavior of the PF scheduler.

Average of average throughputs. A simple scheme is to choose the average of average throughputs of all existing users in this cell as the initial average throughput of the new user.

Minimum of average throughputs. Since new users often join a cell from the edge of the cell, they are expected to have the poorest channel condition. Therefore, this scheme chooses the minimum of average throughputs of all existing users as the initial average throughput of the new user.

Determined by the user. Finally, since users perform tasks such as measuring channel quality and pilot for multiple cells, an intuitive scheme is to let users report their initial average throughput.

Attack results. Fig. 1a shows the fraction of time slots that the attackers acquired where there was one attacker per cell of 50 users and the attackers determined their initial throughput. It shows that after about 2,000 time slots, the attackers consistently obtained about 78 percent of all the slots, a condition that we call the stabilization of the attack. In simulating different number of attackers per



(a)

(b)

Fig. 1. Attack on PF where the attackers know the victims' CQIs. (a) Fraction of time slots acquired by two attackers, one per cell of 50 users. (b) Time slots occupied by the attackers in 36 seconds (18,072 time slots). The three lines represent different schemes for assigning the initial average throughput by the base station.

cell and different schemes for assigning the initial average throughput, we noticed that the attack stabilized well before 36 seconds.

Fig. 1b shows the total number of time slots that the attackers obtained in 36 seconds. Unsurprisingly, the more attackers per cell, the more time slots they can obtained. However, even with just one attacker per cell, the attackers obtained from 13,459 (74 percent) to 16,241 (90 percent) time slots, depending on the scheme by which the scheduler assigns the initial average throughput. Among the three schemes, the scheme that let the user provide this initial value is the most vulnerable, where one attacker obtained 16,241 (90 percent) time slots, while five attackers obtained 17,317 (96 percent) time slots.

4.2.3 Realistic Intercell Attack

In simulations presented above, attackers knew at every time slot all other users' channel conditions and their average throughputs. In practice, however, attackers need on-the-fly adjustment of the estimated maximum CQI-to-throughput ratio of all the victim users. This is because each user's average throughput, in every time slot, will increase by $\alpha * CQI$ if he is scheduled, and decrease by a factor of $(1-\alpha)$ otherwise. We propose the following scheme for adjusting the maximum ratio estimation.

Let $c(t)$ be the estimated maximum CQI-to-throughput ratio at time t and $R_i(t)$ be the average throughput of user i at time t. If the attacker is scheduled at time t, the average throughput of all the other users will decrease, $R_i(t) = (1-\alpha) * R_i(t-1)$. Since $c(t)$ estimates the largest $R_i(t)$ of all the victim users, it increases at the same rate, $c(t+1) = c(t)/(1-\alpha)$. When the attacker is not scheduled, on the other hand, only the average rate of the victim user who is scheduled will increase. Therefore,

$$\begin{aligned} c(t+1) &= \max_i \frac{CQI_i(t+1)}{R_i(t)} \\ &\approx \max_i \frac{CQI_i(t+1)}{R_i(t-1)(1-\alpha) + \frac{\alpha}{N} \cdot CQI_i(t)} \\ &= \max_i \frac{\frac{CQI_i(t+1)}{R_i(t-1)}}{(1-\alpha) + \frac{\alpha}{N} \cdot \frac{CQI_i(t)}{R_i(t-1)}} \\ &\approx \frac{c(t)}{(1-\alpha) + \frac{\alpha}{N} \cdot c(t)}. \end{aligned} \tag{10}$$

Some approximations are involved in the above estimation. First, on average, a victim user gets scheduled once every N times when the attacker is not scheduled. Therefore, the average rate of a victim user will increase by $\alpha * CQI_i(t)/N$ approximately when the attacker is not scheduled. Second, when a user is scheduled, his CQI-to-throughput ratio is the maximum among all users. Thus, its value of $CQI_i(t)/R_i(t-1)$ is approximately $c(t)$. Equation (10) summarizes our analysis:

$$c(t+1) = \begin{cases} c(t)/(1-\epsilon), & \text{scheduled,} \\ c(t)/(1+\sigma \cdot (c(t)-1)), & \text{not scheduled,} \end{cases} \tag{11}$$

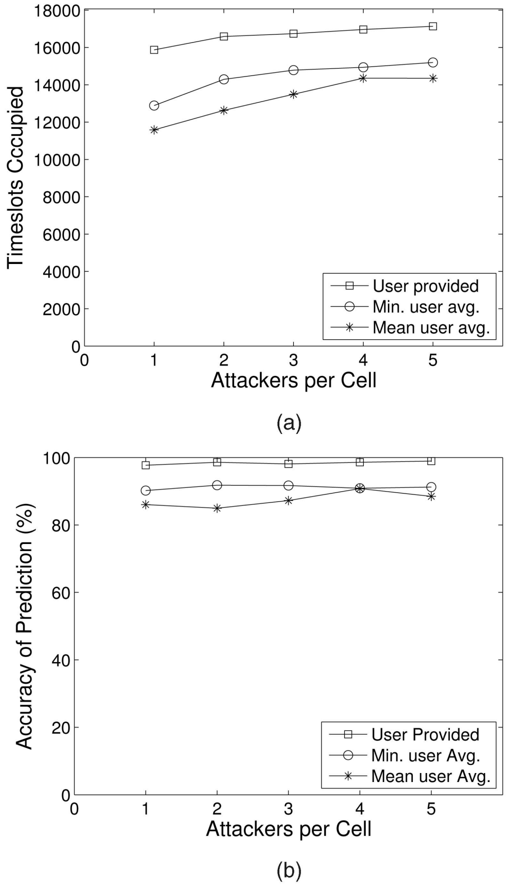
where ϵ and σ are the functions of α. We used ϵ and σ instead of α to compensate for the possible errors in our estimation of the maximum CQI-to-throughput ratio, and determined them empirically.

Attack results. Fig. 2a shows the number of time slots obtained using our prediction strategy. When there is a single attacker per cell of 50 users, the attackers (one in each cell) may obtain between 11,583 (64 percent) and 15,874 (88 percent) time slots, depending on the scheme for assigning initial average throughput. When there are five attackers per cell, they can obtain between 14,353 (79 percent) and 17,136 (95 percent) time slots. Next, Fig. 2b illustrates the accuracy of our prediction scheme simulation. Regardless of the initial average throughput the scheme used, the attackers could always obtain more than 85 percent of the time slots that they would obtain in the ideal simulation (where attackers know the CQIs of the victims). Also, notice the high accuracy of the prediction scheme if the PF scheduler's initial average throughput is user-provided.

Attack impact on throughput. In Fig. 3, before the attack, users' downlink speeds are in range of 40-55 kbps. By comparison, during the attack with a single attacker, each victim's downlink speed has dwindled to 10-15 kbps, while the attacker enjoys almost 1.5 Mbps.

Attack impact on average delay. In Fig. 4, before the attack, the average interpacket transmission delay is about 0.01 s. During the attack, the delay increases to 0.81 s with one attacker present and to 1.8 s with five attackers present.

Attack impact on VoIP. As an example of the attack's impact on delay-sensitive applications, we examined VoIP, which cellular providers now offer. VoIP imposes a rigorous requirement on packet delay: 0-150 ms delay is acceptable,



(a)

(b)

Fig. 2. Attack on PF without knowing victims' CQIs. Each subfigure shows three curves, each representing a different scheme for assigning the initial average throughput. (a) Time slots acquired by the attackers in 18,072 slots. (b) Fraction of time slots acquired by the attackers without knowing the victims' CQIs compared to those with knowing victims' CQIs.

150-400 ms delay might be tolerable, but any delay over 400 ms is disruptive [10]. Additionally, streaming clients employ play out buffers where stream packets are initially buffered in anticipation of expected network delay and possible jitter. This delay budget is end-to-end, including the uplink delay from the sender (U), the transmission delay over the Internet (T, at least 100 ms across the continental USA), the downlink delay to the user (D), and other processing delays for VoIP (O, about 101 ms) [37]. As one attacker can cause 0.81 s downlink delay for victim users, the end-to-end VoIP application delay of all (victim) users in the attacker's cell is at least $T + D + O = 0.10 + 0.81 + 0.10 = 1.01$ s. If five attackers colluded, the end-to-end delay on users' VoIP applications increases to $0.10 + 1.80 + 0.10 = 2.01$ s. Generally speaking, the quality of a VoIP transmission is unaffected if the total packet loss is less than 5 percent of total packets sent [38]. Therefore, excessive delay caused by the attack would render VoIP services useless as undelivered packets dropped during the attack could not be recovered by typical packet-loss-concealment techniques. Additionally, constant buffer underflows would undermine the role of play out buffers as well. As a comparison, geostationary satellite latency is only between 0.24-0.28 ms, which is between 4-8 times shorter than the delay caused by the attack.

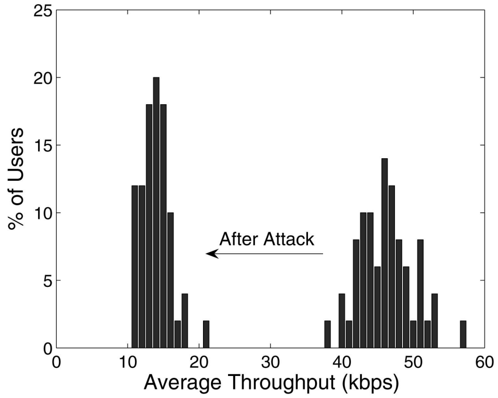


Fig. 3. The impact of PF attack, on average, throughput. The figures on the left and right show the distribution of average throughput of all the users after and before the attack, respectively.

4.3 Temporal Fair Attacks

The essence of the TF attack is similar to that of the PF. When the attack starts, the attacker reports a CQI high enough to obtain the next several time slots. During this process, the offset value of the user keeps decreasing. Once the offset value is low enough such that the user cannot report higher CQI to obtain further time slots, the attackers hand off to an adjacent cell. To elaborate, TF schedules users according to $Q(\vec{U}) = \arg\max_i(U_i + v_i)$, where v_is are nonnegative Lagrange Multipliers associated with the constraints of each user. We used stochastic approximation [8] to calculate v_is. Each user starts with $v_i = 0$. Then, in each time slot, we calculate the next

$$v_i(t+1) = v_i(t) + \frac{1}{t}\left(I_{\{Q(\vec{U}(t))=i\}} - r_i\right),$$

where $I_{\{Q(\vec{U}(t))=i\}}$ is an indicator function such that $I_{\{Q(\vec{U}(t)=i)\}} = 1$ if user i is selected for scheduling, and 0 otherwise. Recall that r_i is the minimum percentage of time when the user i should be able to utilize the spectrum. Therefore, the offset value decreases as the attacker obtains more time slots.

4.3.1 Collusion Strategies

Multiple attackers could use the same collusion strategies as with the PF attack.

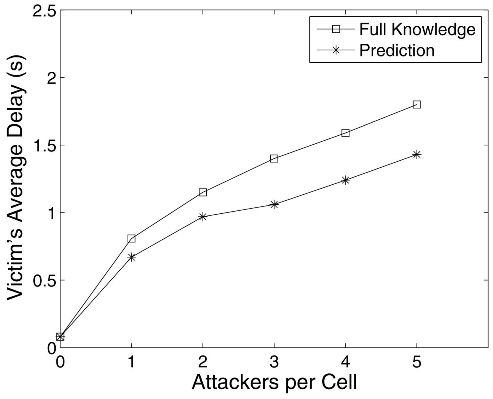
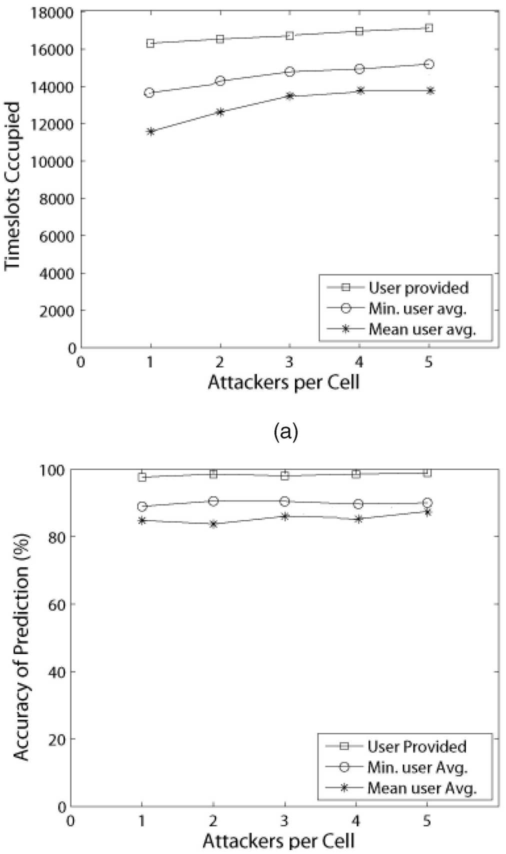


Fig. 4. Average delay of victim users during the attack on PF.



(a)

(b)

Fig. 5. Attack on TF without knowing victims' CQIs. Each subfigure shows three curves, each representing a different scheme for assigning the initial average throughput. (a) Time slots acquired by the attackers in 18,072 slots. (b) Fraction of time slots acquired by attackers without knowing victims' CQIs compared to those with knowing victims' CQIs.

Sequential attack. Attacker with the smallest average throughput $R_i(t)$ starts the attack and obtains as many consecutive time slots as possible, while the other attackers lurk. When the active attacker's offset gets too large, it stops the attack, while the attacker with the smallest average throughput takes over the attack.

Minimum CQI attack. At each time slot, each attacker, given its current average data rate, computes the minimum CQI that it needs to obtain the largest offset. The attacker with the smallest computed CQI becomes the current attacker.

Delta CQI attack. At each time slot t, each attacker i computes $\delta_i(t) = |CQI_i(t) - CQI_i(t-1)|$. The attacker with the smallest $\delta_i(t)$ becomes the current attacker.

Attack results. Again, we ran simulations for 18,072 time slots, or about 36 seconds, in cells of 50 users. We repeated each simulation 100 times. We reused all the nonscheduler-related parameters for PF and changed only the scheduling algorithm to TF. In all our experiments, attacks stabilized well before the end of the run. Additionally, we only ran the most realistic attack where attackers did not know victim users' CQIs, and therefore, had to adjust their offsets on-the-fly.

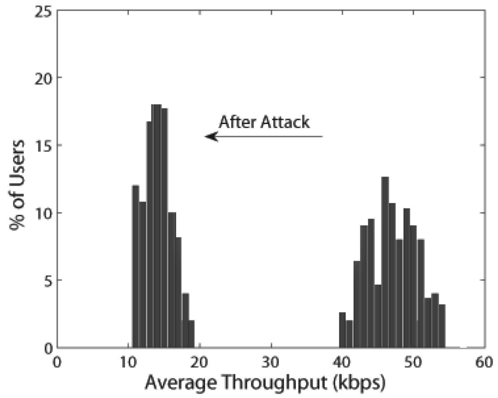


Fig. 6. The impact of TF attack, on average, throughput. The figures on the left and right show the distribution of average throughput of all the users after and before the attack, respectively.

As expected, intracell attacks again prove to be ineffective—One attacker captured 17, while five attackers captured 83 consecutive time slots. However, during intercell attacks, five colluding attackers always captured more than 81 percent of time slots (Fig. 5a). Unsurprisingly, the best colluding scheme was user-provided initial offset where five colluding attackers captured 94 percent of time slots. Fig. 5b shows that the attackers without knowing the victims' CQIs could still obtain more than 80 percent of the time slots that they would obtain when knowing the victims' CQIs. Fig. 6 shows the attack's impact on the average throughput of victim users, and Fig. 7 shows the average delay of victim users during the attack.

5 Defense Strategies

To defend against attacks on opportunistic schedulers, we first evaluate a set of variations of the PF and TF scheduler. Then, we propose a new handoff scheme that can effectively prevent the attacks.

5.1 Scheduler Modifications

We have discussed the pure PF and TF schedulers so far. There are, however, variations of the PF (TF) scheduler, known as hybrid PF (TF) schedulers. These hybrid PF (TF) schedulers were proposed for Quality of Service (QoS). Here, we examine how resilient they are against the attacks discussed in previous sections.

5.1.1 Priority Queue

The base station can use priority queues to alleviate the impact of attacks outlined in the previous section. In

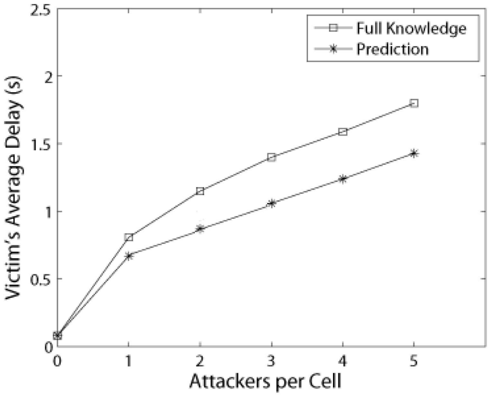


Fig. 7. Average delay of victim users during the attack on TF.

particular, the base station can schedule traffic with delay constraints, such as VoIP traffic, with high priority while scheduling other traffic, such as Web browsing, with low priority. For instance, the scheduler can update a priority scheduling candidate set with devices that have VoIP packets buffered at the base station, that have pending retransmissions in their HARQ manager, or whose head-of-line packet delay is greater than some value [40]. Because the number of high-priority users is relatively small, these users have much better delay performance. Thus, the effect of the attack will be mitigated. The actual impact of priority queue depends on the extent of system manipulation by the attacker. For instance, an attacker may opt out of the priority set if he needs to stay dormant to lower his average throughput value. He can achieve this by keeping his buffer at the base station empty or by reporting extremely low CQI values. During the attack, the attacker can opt into the priority set through one of the following methods: masquerading as a high-priority user (such as a VoIP user), triggering fast retransmissions, and underflowing the buffer at the base station (if the queue length is considered in scheduling decisions).

5.1.2 Round-Robin

Typically, system designers have to balance trade-offs between short-term performance and overall throughput. To improve the delay performance (i.e., a form of short-term fairness), any scheduler can be combined with round-robin scheduler but with additional constraints such that each user should get scheduled for m TTIs within a certain time window w, where $m \leq w/N$ and N is the number of users to be scheduled. Long-term PF fairness, on the other hand, guarantees that each user obtains roughly the same number of time slots over a long period of time (usually, during the lifetime of a user—on the order of minutes). Long-term TF fairness guarantees that each user achieves his predetermined share of time slots. Choosing a smaller w and a larger m improves short-term performance at the expense of lowering the overall throughput. Conversely, choosing a larger w and smaller m improves overall throughput, as the scheduler has more flexibility in choosing a user with good channel conditions, at the expense of short-term performance. In both these cases, the impact of the attacker can be largely mitigated because the attacker cannot obtain more than one time slot within a time window of w.

5.1.3 Other Hybrid Schedulers

Besides round-robin, there are other schemes that improve the delay performance. For example, the exponential queue scheduler takes the head-of-line packet delay into scheduling decision [41]. The impact of short-term constraints, on the other hand, depends on the schemes and parameters used. Attack resilience increases as the short-term constraint becomes more restrictive, but the overall throughput decreases because the scheduler has to schedule users given the short-term constraint.

5.2 Robust Handoff Scheme

Section 4 shows that the intracell attack is not effective, because the attackers cannot acquire time slots perpetually. However, attackers can exploit the handoff procedure to launch a much more effective, intercell attack. We now propose a robust handoff scheme to prevent such exploitation. We focus the discussion on the PF scheduler although it applies to TF as well.

Consider a scenario where a user moves from cell A to cell B. The base stations covering these two cells can negotiate an initial average throughput value for the user in cell B. The optimal initial value in cell B may not necessarily be the average value in cell A. This initial value will affect both security and system performance. It should be set high enough to remove the advantage of an oft-handoff attacker, but not too high to cause excessive delays on benign users. In terms of system performance, this value should be set to provide a smooth transition between two cells such that the handoff user will neither acquire nor lose advantage over the current users in the new cell. Finally, to be fair, this value should reflect the transient behavior of the handoff user.

Consider the special case where the *relative* channel fluctuations of users are statistically identical and independent. This assumption roughly holds when users experience Rayleigh fading and the achievable rate is linear to the channel condition. In this special case where relative channel fluctuations depend only on small-scale fading, such as scattering, the fading environment is often statistically identical for all users in a cell. For example, in an urban environment, users experience rich scattering, and thus, Rayleigh fading. Note that users can have different *average* channel conditions, e.g., depending on their distance to the base station.

When users experience statistically identical and independent relative channel fluctuations, multiuser diversity gain depends only on the number of users in a cell and the statistics of the channel fluctuation [42]. Assuming stationarity and ergodicity, the expectation of the average user throughput $E(R)$ can be expressed as

$$E(R) = E(CQI)\frac{G(N)}{N}, \tag{12}$$

where CQI is a random variable representing the user's channel condition, N is the number of cell users, $E(CQI)/N$ is the average throughput of the user without opportunistic scheduling, and $G(N)$ is the opportunistic scheduling gain.

We propose the following heuristic to set the initial value of the handoff user. Consider that a user moves from cell A to cell B. Let CQI_A and CQI_B represent the channel condition of the user in cells A and B, respectively. Note that CQI_A and CQI_B are random variables. Let N_A and N_B be the number of users in cells A and B, respectively. Let R_A be the current average rate of the user before handoff. The initial value after handoff, R_B^{init}, is set as

$$\begin{aligned} R_B^{init} &= \frac{R_A(i)}{E(CQI_A)\frac{G(N_A)}{N_A}} E(CQI_B)\frac{G(N_B)}{N_B} \cdot (1-\alpha) \\ &= \frac{R_A(i)}{E(R_A)} E(R_B) \cdot \left(1 - \frac{1}{t_c}\right), \end{aligned} \tag{13}$$

where $E(CQI_A)\frac{G(N_A)}{N_A}$ is the expected rate of the user in cell A (following (11)),

$$E(CQI_B)\frac{G(N_B)}{N_B}$$

TABLE 1
Percentage of Time Slots Acquired by Attackers with and without Robust Handoff Procedure

N_A, N_B are the number of users in $Cell_A$ and $Cell_B$, respectively. M is the number of attackers per cell.

	Without robust handoff[1]				With robust handoff			
Scheduler	PF		TF		PF		TF	
Number of attackers per cell	$M = 1$	$M = 5$	$M = 1$	$M = 5$	$M = 1$	$M = 5$	$M = 1$	$M = 5$
$N_A = 10$ ($N_B = 10$)	92.4	94.8	91.9	93.8	11.2	50.8	12.8	53.1
$N_A = 10$ ($N_B = 50$)	92.6	94.8	91.9	94.4	11.5	51.2	12.9	54.6
$N_A = 50$ ($N_B = 10$)	92.6	94.8	91.9	94.4	2.1	10.8	2.7	11.7
$N_A = 50$ ($N_B = 50$)	92.6	94.8	91.9	94.4	2.4	11.0	2.6	11.9

[1] The data in this column were collected in a more accurate simulation environment than those reported in Section 4.2.3. Each data in this table represents the average of 10000 simulation runs, while the data in Section 4.2.3 are averages of 100 runs.

is the expected rate of the user in cell B, and t_c is the time window over which the average rate is calculated. In developing this formula, one should not set the initial value after handoff to be too high to disadvantage the user; the value needs to be just high enough to deter attacks. For example, setting the initial value naively to

$$R_B^{init} = \frac{R_A}{E(R_A)} E(R_B) \cdot \left(1 - \frac{1}{t_c}\right) \tag{14}$$

may cause benign users unjustifiable delays. Typically, the ratio for a benign user is determined by whether the user is in a favorable (with respect to its expectation) or a hindering position. This fact is taken into consideration in the handoff procedure for fairness. In general, we expect $R_A \approx E(CQI_A)\frac{G(N_A)}{N_A}$. We also note that $E(R_B^{init}) = E(R_B)$, which indicates that the user is in a fair position in the new cell. In other words, (13) is an unbiased estimation for the value of R_B^{init}.

In practice, the values of R_A, N_A, and N_B are known. The values of $G(\cdot)$, CQI_A, and CQI_B can be estimated. During an attack, an attacker may manipulate its CQI_B to acquire unfair advantages. Because often a user hands off to a new cell with stronger signal strength, $E(CQI_B) \geq E(CQI_A)$. On the other hand, $E(CQI_B)$ should not be significantly higher than $E(CQI_A)$, because otherwise, the handoff should have been initiated earlier. Additionally, $G(N_A) \approx log(N_A)$ and $G(N_B) \approx log(N_B)$. Therefore, to deter attackers and avoid estimating CQI_A and CQI_B, we can set

$$R_B^{init} \approx \frac{R_A(i)}{\frac{log(N_A)}{N_A}} \frac{log(N_B)}{N_B} \cdot \left(1 - \frac{1}{t_c}\right). \tag{15}$$

We use a similar strategy to defend against the attack on the TF scheduler. While the PF defense adjusts the average data rate during handoff, the TF defense adjusts the offset using the following formula:

$$v_B^{init} \approx \frac{v_A(i)}{\frac{log(N_A)}{N_A}} \frac{log(N_B)}{N_B} \cdot \left(1 - \frac{1}{t_c}\right). \tag{16}$$

5.2.1 Evaluation of the Robust Scheduler

To examine the effectiveness of our robust handoff procedure, we ran simulations on two cells (A and B) with varying number of users (N_A and N_B for cells A and B, respectively) and varying number of attackers per cell (M). For simplicity, we assume that each cell has the same number of attackers. Table 1 summarizes the results. For example, when there is one attacker per cell and $N_A = 10$ and $N_B = 50$, he or she can acquire 92 percent time slots in cell A. Whereas with our robust handoff procedure, the attacker can acquire only 11.5 percent time slots, which not only is substantially lower, but also is close to the long-term fairness goal of the PF scheduler (10 percent). In simulating the same robust handoff procedure on TF using the same parameters, the attackers acquired only 12.9 percent time slots in cell A. As another example, when there are five attackers per cell and $N_A = 10$ and $N_B = 50$, without the robust handoff procedure, the attackers on PF acquired 94.8 percent time slots, whereas with the robust handoff procedure, the attackers acquired only 51.2 percent time slots—again close to the long-term fairness goal of PF (50 percent time slots). With the robust handoff procedure on TF, the attackers acquired about 54.6 percent time slots.

6 Related Work

Studies on the security of 3G networks began to appear in recent years [43], [44], [45]. Sridharan et al. modeled the uplink channel in EV-DO and suggested that malicious users could modify their power transmission levels to cause interference on honest users [46]. Our work, on the other hand, concentrates on the downlink due to its considerably higher bandwidth. Furthermore, we also develop attacks as well as provide defense.

As resources on cellular networks are much more limited than those on the Internet, Denial of service (DoS) attacks on cellular networks have risen to prominence. Agarwal et al. [47] conducted a capacity analysis of shared control channels used for SMS delivery. They concluded that increasing volume and message sizes can significantly affect network performance. Enck et al. [48] introduced a DoS attack by sending a sufficient rate of SMS messages to a range of local cell phones. Traynor et al. [49] evaluated this attack using in a GSM simulator and proposed mitigation strategies. Furthermore, Traynor et al. pointed out that despite efforts to securely overlay a packet-switched network onto a circuit-switched network, connection establishment is still vulnerable to low-bandwidth DoS attacks [50]. Racic et al. [31] showed that attackers can deplete cellular phones' batteries up to 22 times faster by exploiting Multimedia Messaging Service (MMS) and data packet services in the cellular network. All these studies focused on attacks originating from outside the cellular network, usually from the Internet. In contrast, this work focuses on DoS attacks from inside the cellular network using existing mobile devices, such as cellular phones and 3G cards.

Significant amount of research has been conducted on efficient resource sharing in cellular networks. In particular, researchers in [8], [23], [51], [52] extensively studied opportunistic scheduling algorithms. However, prior work focused on improving system performance under various system constraints and requirements, including the effect on TCP performance [24], [25], instability [26], and multicell scheduling [29]. Moreover, Vukadinovic and Karlsson [53] argue strongly that opportunistic schedulers, and PF in particular, may not provide users with the best performance when streaming flows constitute a significant share of the traffic load. In contrast to these studies, we consider the threat of malicious users and their impact. While (artificial) handoff has been considered for load-balancing in [29], [54], we propose a method for assigning good initial values for security.

Concurrently with and independently of our work, Bali et al. have shown that a sudden arrival of packets to a malicious mobile device whose buffer had been empty for a period of time can starve constant network flow of a victim mobile device [6]. The authors experimented on an isolated EV-DO network testbed using two devices. Their exploit indirectly influenced the PF scheduler by sending bursty traffic. By contrast, our attack directly manipulates the PF scheduler by sending fake CQI reports, which has much bigger impact. Our attack outperformed theirs in a simulation under the same network condition. In a cell with only two users, our attack occupied 1,198 consecutive slots, while their method occupied only 529 slots. In a cell with 50 users, our method occupied 65 slots, while theirs occupied only 20 slots. Moreover, since their attack exploits the fact that a user's average rate drops when his or hers buffer is empty, the network can mitigate their attack by limiting the decrease of average throughput when the buffer is empty [28]. Finally, we also discovered a vulnerability in the handoff procedure that, in addition to the vulnerability of fake CQI reports, could perpetually attack PF and TF scheduler. Furthermore, we proposed a robust handoff algorithm that mitigates the attack.

7 Conclusion

We have shown that cellular data networks are vulnerable to DoS attacks by malicious mobile devices because of the following vulnerabilities:

- The network trusts mobile devices to report CQIs, which the PF and TF schedulers use without verification for assigning time slots. However, malicious mobile devices can manipulate their reported CQIs to gain a large number of time slots.
- The network does not track the average throughput of mobile devices across different cells. Therefore, malicious devices can maintain perpetual scheduling priority by frequent handoffs.

Our simulations show that just one attacker per cell can decrease the throughput and increase the delay of victim users significantly enough to disrupt time-sensitive data services, such as VoIP. Moreover, multiple attackers can collaborate to aggravate the attack. To defend against the attacks, we discuss modifications to the PF and TF schedulers, and propose a robust handoff procedure. Simulations show that our robust handoff procedure effectively enforces long-term fairness and prevents the attacks.

Acknowledgments

This work was partially supported by the US National Science Foundation (NSF) under grant CNS 0644450 and by a generous gift from Sprint. The authors thank Jean Bolot, Sridhar Machiraju, Tao Ye, and Hui Zang for their valuable comments.

References

[1] H. Holma and A. Toskala, *HSDPA/HSUPA for UMTS.* John Wiley & Sons, 2006.

[2] V. Vanghi, A. Damnjanovic, and B. Vojcic, *The CDMA2000 System for Mobile Communications.* Prentice Hall, 2004.

[3] 3GPP, "3G Wlan—Trust Model," http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_25_Munich/Docs/PDF/S3-020523.pdf, 2008.

[4] Ericsson, "WCDMA Evolved—The First Step—HSDPA," http://www.ericsson.com/technology/whitepapers/wcdma_evolved.pdf, 2008.

[5] Qualcomm, "HSDPA for Improved Downlink Data Transfer," Oct. 2004.

[6] S. Bali, S. Machiraju, H. Zang, and V. Frost, "On the Performance Implications of Proportional Fairness (PF) in 3G Wireless Networks," *Proc. Passive and Active Measurements Conf.,* 2007.

[7] S.Z. Asif, "Aligning Business and Technology Strategies—An Evolution of a Third Generation Wireless Technology," *Proc. Eng. Management Conf.,* 2002.

[8] X. Liu, E.K.P. Chong, and N.B. Shroff, "Opportunistic Transmission Scheduling with Resource-Sharing Constraints in Wireless Networks," *IEEE J. Selected Area in Comm.,* vol. 19, no. 10, pp. 2053-2064, Oct. 2001.

[9] S.S. Kulkarni and C. Rosenberg, "Opportunistic Scheduling Policies for Wireless Systems with Short Term Fairness Constraints," *Proc. IEEE Global Telecomm. Conf. (Globecom),* 2003.

[10] ITU-T, "One-Way Transmission Time," ITU-T Recommendation G.114, 1996.

[11] G. Varrall and R. Belcher, *3G Handset and Network Design.* Wiley and Sons, 2004.

[12] *HSDPA Mobile Broadband Data,* Agere Systems, 2005.

[13] M. Grosslauser and D. Tse, "Mobility Increases the Capacity of Wireless Ad Hoc Networks," *Proc. IEEE INFOCOM,* Apr. 2001.

[14] A. Jalali, R. Padovani, and R. Pankaj, "Data Throughput of CDMA-HDR a High Efficiency-High Data Rate Personal Communication Wireless System," *Proc. IEEE Vehicular Technology Conf.,* vol. 3, 2000.

[15] E.F. Chaponniere, P. Black, J.M. Holtzman, and D. Tse, *Transmitter Directed Multiple Receiver System Using Path Diversity to Equitably Maximize Throughput,* US Patent No. 6449490, 2002.

[16] H. Kushner and P. Whiting, "Convergence of Proportional-Fair Sharing Algorithms under General Conditions," *IEEE Trans. Wireless Comm.,* vol. 3, pp. 1250-1259, 2004.

[17] P. Rysavy, "Data Capabilities: GPRS to HSDPA and BEYOND," http://www.cingular.com/b2b/content/downloads/DataCapabilities_beyond.pdf, 2009.

[18] T.E. Kolding, "Link and System Performance Aspects of Proportional Fair Scheduling in WCDMA/HSDPA," *Proc. Vehicular Technology Conf.,* 2003.

[19] S. Nanda, K. Balachandran, and S. Kumar, "Adaptation Techniques in Wireless Packet Data Services," *IEEE Comm. Magazine,* vol. 38, no. 1, pp. 54-64, Jan. 2000.

[20] 3GPP, "Radio Resource Control (RRC): Protocol Specification, 3GPP TS 25.331 Version 7.00," http://3gpp.org/ftp/Specs/html-info/25331.htm, 2009.

[21] 3GPP, "Physical Layer Procedures: Protocol Specification, 3GPP TS 25.214 Version 7.00 Release 7," http://3gpp.org/ftp/Specs/html-info/25214.htm, 2009.

[22] F. Kelly, "Charging and Rate Control for Elastic Traffic," *European Trans. Telecomm.*, vol. 8, pp. 33-37, 1997.

[23] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*, first ed. Cambridge Univ. Press, 2005.

[24] M. Assaad, B. Jouaber, and D. Zeghlache, "Effect of TCP on UMTS-HSDPA System Performance and Capacity," *Proc. IEEE Global Telecomm. Conf.*, 2004.

[25] J.-H. Choi, J.-G. Choi, and C. Yoo, "Analyzing the Impact of Proportional Fair Scheduler on TCP Performance," *Proc. Fourth IFIP Int'l Conf. Network Control and Eng. for QoS, Security, and Mobility*, 2005.

[26] M. Andrews, "Instability of the pf Scheduling Algorithm for HDR," *IEEE Trans. Wireless Comm.*, vol. 3, no. 5, pp. 1422-1426, Sept. 2004.

[27] T.E. Kolding, "Link and System Performance Aspects of Proportional Fair Scheduling in WCDMA/HSDPA," *Proc. Vehicular Technology Conf. (VTC)*, 2003.

[28] J. Yang, Z. Yifan, W. Ying, and Z. Ping, "Average Rate Update Mechanism in PF Scheduler in HDR," *Proc. Global Telecomm. Conf. (Globecom)*, 2004.

[29] T. Bu, L. Li, and R. Ramjee, "Generalized PF Scheduling in Third Generation Wireless Data Networks," *Proc. IEEE INFOCOM*, 2006.

[30] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi, "CDMA/HDR: A Bandwidth-Efficient High-Speed Wireless Data Service for Nomadic Users," *IEEE Comm. Magazine*, vol. 38, no. 7, pp. 70-77, July 2000.

[31] R. Racic, D. Ma, and H. Chen, "Exploiting MMS Vulnerabilities to Stealthily Exhaust Mobile Phones' Battery," *Proc. IEEE SecureComm*, 2006.

[32] "Homebrew Mobile Phone Club," http://telefono.revejo.org, 2008.

[33] D. Ilett and M. Hines "Skulls Program Carries Cabir Worm into Phones," http://news.com.com/Skulls+program+carries+Cabir+worm+into+phones/2100-7349_3-5469691.html, 2009.

[34] Novatel, "Novatel Merlin u870 PC Card," http://www.novatelwireless.com/products/merlin/merlin-u870.html, 2008.

[35] K. Tuna, "Hacking EVDO," *Proc. Defcon 15*, 2007.

[36] H. Suzuki, "Statistical Model for Urban Radio Propagation," *IEEE Trans. Communications*, vol. COM-25, no. 7, pp. 673-680, July 1977.

[37] B. Goode, "Voice over Internet Protocol (VoIP)," *Proc. IEEE*, vol. 90, no. 9, pp. 1495-1517, Sept. 2002.

[38] H. Zheng, G. Rittenhouse, and M. Recchione, "The Performance of Voice over IP over 3G Downlink Shared Packet Channels under Different Delay Budgets," *Proc. Vehicular Technology Conf. (VTC)*, 2003.

[39] K.I Pedersen, P.E. Mogensen, and T.E. Kolding, "QoS Considerations for HSDPA and Performance Results for Different Services," *Proc. IEEE Vehicular Technology Conf.*, 2006.

[40] B. Wang, K.I. Pedersen, T.E. Kolding, and P.E. Mogesen, "Performance of Voip on HSDPA," *Proc. Vehicular Technology Conf. (VTC)*, 2005.

[41] S. Shakkottai and A. Stolyar, "Scheduling for Multiple Flows Sharing a Time-Varying Channel: The Exponential Rule," *Analytic Methods in Applied Probability*, Am. Math. Soc., 2001.

[42] S. Borst, "User-Level Performance of Channel-Aware Scheduling Algorithms in Wireless Data Networks," *Proc. IEEE INFOCOM*, 2003.

[43] K. Kotapati, P. Liu, Y. Sun, and T.F.L. Porta, "A Taxonomy of Cyber Attacks on 3G Networks," Technical Report NAS-TR-0021-2005, Network and Security Research Center, Dept. of Computer Science and Engineering, Pennsylvania State Univ., 2005.

[44] F. Ricciato, "Unwanted Traffic in 3G Networks," *ACM SIGCOMM Computer Comm. Rev.*, vol. 36, no. 2, 2006.

[45] A. Bovosa, "Attacks and Counter Measures in 2.5G and 3G Cellular IP Networks," Juniper White Paper, 2004.

[46] A. Sridharan, R. Subbaraman, and R. Guerin, "Uplink Scheduling in the EV-DO Rev. A System: An Initial Investigation," Sprint ATL Research Report Nr. RR06-ATL-080139, 2006.

[47] N. Agarwal, L. Chandran-Wadia, and V. Apte, "Capacity Analysis of the GSM Short Message Service," *Proc. Nat'l Conf. Comm.*, 2004.

[48] W. Enck, P. Traynor, P. McDaniel, and T.L. Porta, "Exploiting Open Functionality in SMS-Capable Cellular Networks," *Proc. 12th ACM Conf. Computer and Comm. Security (CCS '05)*, Nov. 2005.

[49] P. Traynor, W. Enck, P. McDaniel, and T.L. Porta, "Mitigating Attacks on Open Functionality in SMS-Capable Cellular Networkss," *Proc. ACM MobiCom*, 2006.

[50] P. Traynor, P. McDaniel, and T.L. Porta, "On Attack Causality in Internet-Connected Cellular Networks," *Proc. USENIX Security Symp. (SECURITY)*, 2007.

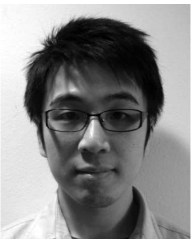
[51] R. Knopp and P. Humblet, "Information Capacity and Power Control in Single-Cell Multiuser Communications," *Proc. Int'l Conf. Comm. (ICC)*, 1995.

[52] X. Liu, E.K.P. Chong, and N.B. Shroff, "A Framework for Opportunistic Scheduling in Wireless Networks," *Computer Networks*, vol. 41, no. 4, pp. 451-474, Mar. 2003.

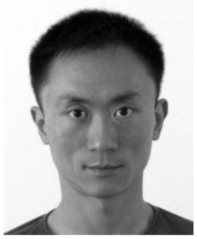
[53] V. Vukadinovic and G. Karlsson, "Video Streaming in 3.5G: On Throughput-Delay Performance of PF Scheduling," *Proc. 14th IEEE Int'l Symp. Modeling, Analysis, and Simulation*, 2007.

[54] A. Sang, X. Wang, M. Madihian, and R.D. Gitlin, "Coordinated Load Balancing, Handoff/Cell-Site Selection, and Scheduling in Multi-Cell Packet Data Systems," *Proc. ACM MobiCom*, pp. 302-314, 2004.

Radmilo Racic received the MS degree in computer science in 2009 from the University of California, Davis, where he is currently working as a security consultant focusing on penetration testing and network security.



Denys Ma received the MS degree in computer science from the University of California, Davis, in 2007. He is currently a researcher at McAfee Avert Labs. He is working on intrusion prevention technologies and network security.



Hao Chen received the PhD degree in computer science from the University of California, Berkeley, in 2004. He is currently an assistant professor at the University of California, Davis. His research focuses on various topics in computer security. He received the US National Science Foundation CAREER Award in 2007. He is a member of the IEEE and the IEEE Computer Society.

Xin Liu received the PhD degree in electrical engineering from Purdue University in 2002. She is currently an associate professor in the Computer Science Department at the University of California, Davis. Her research is on wireless communication networks, with a focus on resource allocation and dynamic spectrum management. She received the US National Science Foundation (NSF) CAREER Award in 2005 for her research on Smart-Radio-Technology-Enabled Opportunistic Spectrum Utilization. She is a member of the IEEE.