# Exploiting Opportunistic Scheduling in Cellular Data Networks

Radmilo Racic, Denys Ma, Hao Chen, Xin Liu
University of California, Davis
{racic,madl,hchen,liu}@cs.ucdavis.edu

## Abstract

*Third Generation (3G) cellular networks utilize time-varying and location-dependent channel conditions to provide broadband services. They employ opportunistic scheduling to efficiently utilize spectrum under fairness or QoS constraints. Opportunistic scheduling algorithms rely on collaboration among all mobile users to achieve their design objectives. However, we demonstrate that rogue cellular devices can exploit vulnerabilities in opportunistic scheduling algorithms, such as Proprotional Fair (PF), to usurp the majority of time slots in 3G networks. Our simulations show that only five rogue device per 50-user cell can use up to 90% of the time slots, and can cause 2 seconds of end-to-end inter-packet transmission delay on VoIP applications for every user in the same cell, rendering VoIP applications useless. To defend against these attacks, we explore several detection and prevention schemes, including modifications to the PF scheduler and a secure handoff procedure.*

## 1 Introduction

The specification for 3G cellular data services recommends implementing an *opportunistic scheduler*. Both HSDPA [18] and EV-DO [44] use an opportunistic scheduler in the downlink to profit from multi-user diversity. Multi-user diversity utilizes fading and shadowing of cellular users within a single cell to optimize bandwidth efficiency [17]. 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 mobile devices to manage network operations [1]. Unfortunately, this principle is often violated, as in the case of opportunistic scheduling in 3G networks. A popular scheduling algorithm is *Proportional Fair* (PF) [4, 6, 14, 18, 31, 44], which maximizes the product of the throughput delivered to all users [11, 14, 21, 24, 34].

We discovered two vulnerabilities in the PF scheduler:

1. The scheduler trusts channel condition reports from mobile devices without verification.
2. The scheduler fails to track a mobile device's average channel condition during handoff.

A malicious mobile device can exploit these vulnerabilities by misrepresenting its channel conditions and initiating unnecessary handoffs (to obtain a fresh average channel condition) to usurp a large number of time slots at the expense of other users. Our simulations show that only one attacker per 50-user cell can occupy between 74% to 90% of all the time slots persistently. To put it in another perspective, when users are running VoIP applications, one attacker per cell can perpetuate a 1 second end-to-end inter-packet 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 [20], this attack would render VoIP useless.

We discuss a variety of modifications to the PF scheduler and their resilience to the attack. However, as the PF scheduler operates within a single cell, it cannot guarantee its goal of long term fairness to mobile devices that can hand off freely across cells. Therefore, we propose a robust handoff procedure to ensure graceful handoff for honest users but at the same time to prevent attackers from usurping bandwidth.

We make the following contributions:

- We identify vulnerabilities in the Proportional Fair scheduler, and analyze a series of attacks mathematically. Our simulations demonstrate that these attacks would devastate victim mobile users by causing persistent delays and lowering throughput.
- We study a variety of modifications to the PF schedulers to scrutinize their resilience to the above-mentioned attacks.
- We propose fortifying the PF scheduler with a robust handoff algorithm to mitigate these attacks.

## 2 Attack overview

3G cellular networks grant unwarranted trust to mobile devices, allowing them to report channel conditions and to initiate handoffs at their discretion. By exploiting these vulnerabilities, malicious mobile devices can usurp a majority of downlink[1] scheduling slots, causing intolerable delays to the victim users and rendering many network services virtually useless. In this section, we will provide an overview of the 3G data network technologies for understanding these vulnerabilities.

### 2.1 3G data networks

With the goal of avoiding major network restructuring, cellular providers have developed two new data services, EV-DO and HSDPA, to provide broadband-like downlink speed for emerging applications, such as Voice-over-IP (VoIP) and streaming video. In both services, the downlink utilizes time division multiplexing (TDM) by dividing the channel in time slots, or Transmission Time Intervals (TTIs). (Note that $TTI_{EV-DO} = 1.67ms$ and $TTI_{HSDPA} = 2ms$.) The *scheduler* at each base station selects a single user[2] to transmit at each TTI. Both services rely on two main techniques to increase efficiency in the downlink direction: *link adaptation* and *fast retransmissions*. In link adaptation, mobile devices report quasi instantaneous downlink channel quality information, *channel quality indicator* (CQI), to base stations. The base station can then adapt data rate contingent on channel conditions: the better the channel condition, the higher the data rate [38]. Fast retransmission (part of the Hybrid Automatic Repeat Request (HARQ) manager) is HSDPA's retransmission mechanism that allows a mobile device to NACK each erroneous downlink packet to request a retransmission from its base station instead of the sending server.

#### 2.1.1 Opportunistic scheduling

Channel conditions of cellular mobile devices are time-varying and location-dependent due to fading and shadowing. This causes the multi-user diversity effect: since many users fade independently, at any given time some subset of users will likely have strong channel conditions. Since instantaneous channel conditions derive the instantaneous data rates of mobile devices [29], mobile devices periodically measure and report their CQIs to their base stations. 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. Thus, opportunistic schedulers often achieve higher network performance than schedulers that do not take into account channel conditions, such as round robin. A very popular opportunistic scheduler is Proportional Fair (PF) [4, 6], whose goal is to maximize the product of the throughput delivered to all users [11, 21].

In PF, each mobile device measures its instantaneous channel conditions through pilot signals[3], 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. To achieve the goal of maximizing the product of the throughput delivered to all users [22], the PF scheduler chooses the user with the highest ratio of $CQI_i(t)/R_i(t)$ where $R_i(t)$ Is the average throughput of user $i$ at time $t$. [4] The base station estimates $R_i(t)$ as follows:

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

where $\alpha$ is a network provider's parameter describing the weight of the current time slot toward the average. A typical $\alpha$ is $0.001$.

While current 3G standards do not select a particular opportunistic schedule, PF is the most popular both in the research community [3, 5, 10, 12, 25, 42, 46] and in industry [4, 6, 7, 14, 18, 31, 44]. Networks may implement modified versions of PF schedulers. For instance, a PF scheduler may apply code multiplexing by scheduling multiple users within the same Transmission Time Interval (TTI). In this case, one TTI may be divided into 15 channels using different channelisation codes [31, 38]. The maximum number of codes that a user could obtain is determined by the mobile device's capability. In each TTI, the PF scheduler selects a single mobile device if the device can receive all the codes; otherwise, the PF scheduler selects multiple devices to share the codes. Researchers have also proposed variations of the PF scheduler, such as combining the PF scheduler with a priority queue or the round robin scheduler. For the rest of the paper, we will refer to the original PF discussed in detail above as the PF scheduler, and will refer to modified PFs as the hybrid PF schedulers.

[1] From the network to the mobile users.

[2] The scheduler may also select a scheduling candidate set of users to be transmitted at each TTI.

[3] A continuous stream of signal sent by the base station to help devices sychronize and measure their signal strength.

[4] PF makes scheduling decisions based on the ratio $DRC_i(t)/R_i(t)$ where $DRC_i(t) = min\{CQI_k[n], \frac{B_k[n]}{t_{TTI}}\}$ and $B_k[n]$ is the buffer size. In this analysis, we opt to eliminate buffer dependence for simplicity.

### 2.1.2 Handoff

Cellular networks implement *handoffs* to transfer a connection 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 handoff to another base station [31].[5] 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.

## 2.2 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 seemingly innocuous actions different from what is intended by the providers, even when providers attempt tamper-proof techniques [6,19,32,39]. For instance, attackers can modify their laptops' 3G PC cards, either through the accompanying SDKs [30] or the device firmware [43], to gain access to the network. By trusting all mobile devices, a system that implements the PF scheduler suffers from at least two vulnerabilities, discussed in the following subsections.

### 2.2.1 Fabricated CQIs

Since 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 to usurp the network bandwidth and disrupt other mobile devices. To illustrate this idea, let's consider a naïve attack with one attacker operating in one cell. A malicious mobile device reports an inflated CQI such that its ratio of CQI 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 Equation 1) until its reported CQI exceeds the range of CQI values.

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 2.1.1, $R_i(t)$ is determined by whether the user is scheduled or not as depicted in equation (1). 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 Equation 1, $R_i(s) = (1-\alpha)^N R_i(s-N) + \alpha CQI$. 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 Equation 2.2.1, we have

$$R_i(s) = \frac{\alpha CQI}{1-(1-\alpha)^N} \approx \frac{CQI}{N} \tag{2}$$

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

$$\begin{aligned} R_i(s-1) = R_i(s+N-1) &= (1-\alpha)^{N-1} R_i(s) \\ &\approx (1-\alpha)^{N-1} \frac{CQI}{N} \end{aligned} \tag{3}$$

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

$$C = \frac{CQI}{R_i(s-1)} \approx \frac{N}{(1-\alpha)^{N-1}} \tag{4}$$

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

$$CQI_{max} \geq \frac{C}{(1-\alpha)^{t_0-1}} R_a(0) \cdot \Pi \tag{5}$$

[5] Note that EV-DO implements mobile device initiated hand-offs instead.

where $\Pi$ is

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

Equation (5) shows that the maximum number of consecutive slots an attacker can obtain ($t_0$) depends on the attacker's beginning average throughput ($R_i(0)$), maximum CQI ($CQI_{max}$), and $\alpha$. Maximum CQI depends on network hardware and $\alpha$ is used to balance the trade off between long-term and short-term performance. Since the $CQI_{max}$ and $\alpha$ are set by the system, they are out of the attacker's control. The attacker does have control over $R_i(0)$, its average throughput at the beginning of the attack. Equation (5) shows that the smaller the value $R_a(0)$, the larger the value $t_0$. Therefore, after each attack session, the attacker needs to *reset* its $R_a(0)$ by reporting lower CQI values for a sufficient period (typically a few seconds[6]). Finally, this model is simplified, assuming all victim users have the same, consistent CQI. When users have time-varying channel conditions, Equation 5 provides an upper bound for estimating $t_0$.

#### 2.2.2 Greedy handoffs

Opportunistic schedulers are oblivious to handoffs that mobile devices experience. For example, when a mobile device performs a handoff to another base station, 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 [10, 46]. By reporting fabricated CQIs, as in the naïve attack illustrated in the section above, a malicious mobile device has to report monotonically increasing CQIs to sustain the attack because its average data rate keeps increasing. Eventually, the attack stops when its reported CQI exceeds the maximum allowable CQI. However, if the malicious device sits in the coverage area of multiple base stations, it 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 3.3). Note that by manipulating its CQI reports, a malicious mobile device can cause its base station to initiate a handoff.

## 3 Attack analysis

### 3.1 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 in the overlapping areas of cells.

We believe 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, minimze the risk of discovery and punishment[7]. 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 [19, 32, 39]. 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 start by considering intra-cell attacks with multiple attackers. Then, we describe an inter-cell attack, which is considerably more effective. Finally, we present a more realistic attack where attackers are unaware of other users's channel conditions in the cell.

### 3.2 Intra-cell attack

Consider a scenario where all the 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 3.4 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 coordinate.

**Sequential attack** The simplest scheme is to attack sequentially. The attacker with the smallest average

[6] This limitation is in relation to our naïve attack scenario. Section 3 discusses heuristics for avoiding this limitation.

[7] While gaining momentum in North America, prepaid data plans are very popular and omnipresent in many parts of the world where one can buy prepaid SIM cards anonymously.

throughput $R_i(t)$ starts the attack and obtains as many consecutive time slots as possible, while the other attackers lurk (by reporting arbitrarily small CQIs to avoid being scheduled). When the active attacker's reported CQI exceeds the maximum value of CQI, 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.

#### 3.2.1 Attack results

To verify the effectiveness of this and other attacks described below, we ran simulations for 18072 time slots, or 30 seconds. Through trial-and-error we determined that 30 seconds is more than enough time to determine an attack's effectiveness as all attacks stabilized well before that time. In simulating single cell attacks, we chose parameters that are recommended by the 3G and HSDPA specification or that are commonly used by cellular networks. The PF scheduler has an $\alpha = .001$. We assume 50 users in a cell. Each user quantized his channel condition into 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 [37] 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 [37]. A simulation with only one attacker present showed that the attacker gained an average of 19 time slots, with a standard deviation of 2.77.

Next, we simulated multiple attackers in the same cell. Again, each user's channel condition was a random variable following a Rayleigh distribution. We varied the number of attackers from one to five and simulated each of the attack schemes in Section 3.2. Figure 1 shows 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.

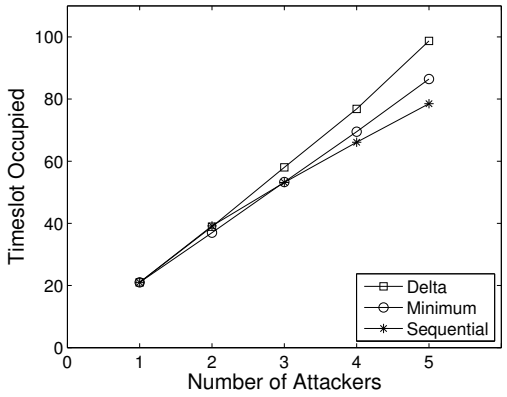



**Figure 1. Consecutive time slots obtained by attackers using different collaborating schemes in a naive single cell attack. Notice that after the attack's initial burst, the attackers must relinquish a large number of timeslots before attacking again**

Although 99 consecutive time slots (or 165ms) 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 2000) of time slots to reset their average throughput low enough before they can attack again. Therefore, this attack is not sustainable. This confirmed our intuition that single cell attacks were relatively ineffective. Fortunately (or unfortunately, depending on your position), we were able to exploit another vulnerability to make our attack much more effective and sustainable.

### 3.3 Inter-cell attack

PF scheduler ensures long-term fairness within a cell's boundary. By transgressing those boundaries, attackers can gain unfair share of network bandwidth. For example, when attackers sit in the overlapping area of two cells, they can exploit the handoff procedure to make their attack much more effective and sustainable. 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 stops. However, since users can

trigger handoffs and the network does not carry users' average throughput across cells, the attacker can handoff to another other cell, get a small initial average throughput, and immediately start the attack in its new cell.

#### 3.3.1 Initial average throughput

Since the network does not track users' average throughput across cells [10], when a new user joins a cell, the scheduler must first assign the user an initial value for its average throughput. Since the choice of this initial value is unspecified, we explore three reasonable schemes. Although these schemes are not all-inclusive, they represent good schemes that lead to predictable behavior of the PF scheduler.

**Based on the average of average throughput of all users** A simple scheme is to choose the average of average throughput of all existing users in this cell as the initial average throughput of the new user, since the new user's channel condition is close to the average channel condition of all existing users.

**Based on the minimum of average throughput of all users** 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 the average throughput of all existing users as the initial average throughput of the new user.

**Determined by the user** Finally, since users are trusted with tasks such as channel quality and pilot measurements for multiple cells, an intuitive scheme is to let users report their initial average throughput. A major problem with this scheme is that an attacker can report a bogus low average throughput to gain unfair advantage in scheduling.

#### 3.3.2 Attack Results

Figure 2 shows the fraction of time slots that the attackers procured where there was one attacker per cell and the attackers determined their initial throughput. It shows that after about 2000 time slots, the attackers consistently obtained about 78% of all the slots, a condition that we call the stabilization of the attack. We simulated different number of attackers per cell and different schemes for assigning the initial average throughput, and in all the simulations the attack stabilized well before 30 seconds.

Figure 3 shows the total number of time slots that the attackers obtained in 30 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 13459 (74%) to 16241 (90%) 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 16241 (90%) time slots while five attackers obtained 17317 (96%) time slots.

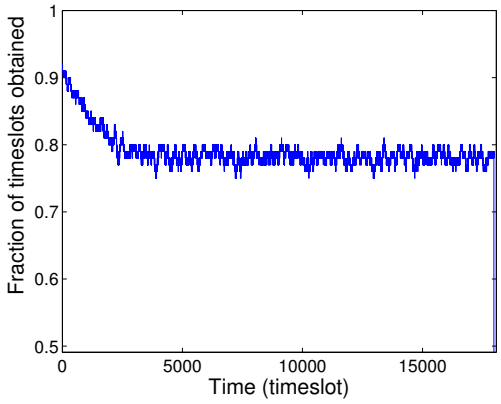



**Figure 2. Fraction of time slots obtained by two attackers, one per cell of 50 users**

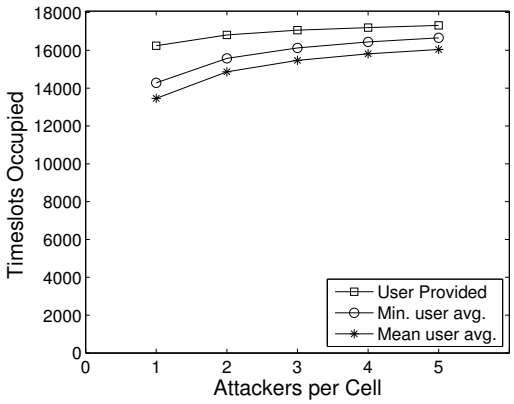



**Figure 3. Time slots occupied by the attack in 30 seconds (18072 time slots). The three lines represent different schemes for assigning the initial average throughput by the base station**

### 3.4 Realistic inter-cell attack

In the above simulations, the attack required attackers to know all users' channel conditions and average throughput at each time slot. In practice, however, attackers may not have such information. In this case, during the attack, the attacker must constantly adjust

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,

$$\begin{aligned} c(t+1) = \max_i \frac{CQI_i(t+1)}{R_i(t)} &\approx \\ \approx \max_i \frac{CQI_i(t+1)}{R_i(t-1)(1-\alpha) + \frac{\alpha}{N} \cdot CQI_i(t)} &= \\ = \max_i \frac{\frac{CQI_i(t+1)}{R_i(t-1)}}{(1-\alpha) + \frac{\alpha}{N} \cdot \frac{CQI_i(t+1)}{R_i(t-1)}} &\approx \\ \approx \frac{c(t)}{(1-\alpha) + \frac{\alpha}{N} \cdot c(t)} & \end{aligned} \quad (6)$$

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 $\alpha/N * CQI_i(t)$ approximately when the attacker is not scheduled. Second, when a user is scheduled, his CQI-to-throughput ratio is the maximum among all user. Thus its value of $CQI_i(t)/R_i(t-1)$ is approximately $c(t)$. Equation 7 summarizes our analysis:

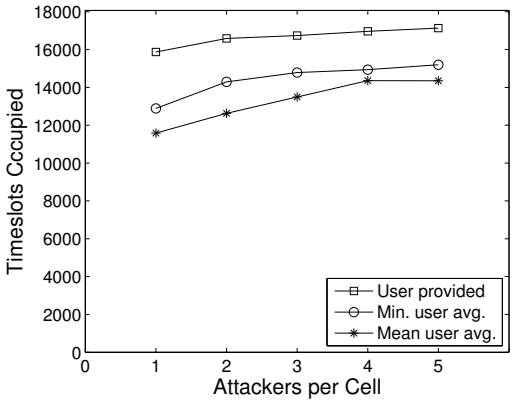
$$c(t+1) = \begin{cases} c(t)/(1-\epsilon) & \text{scheduled} \\ c(t)/(1+\sigma \cdot (c(t)-1)) & \text{not scheduled} \end{cases} \quad (7)$$

where $\epsilon$ and $\sigma$ are functions of $\alpha$. We used $\epsilon$ and $\sigma$ instead of $\alpha$ to compensate for the possible errors in our estimation of the maximum CQI-to-throughput ratio, and determined them empirically.

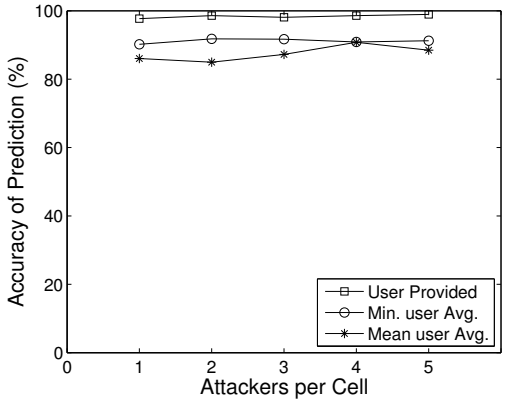
### 3.4.1 Attack results

Figure 4(a) shows the number of time slots obtained using our prediction strategy in simulation. When there is a single attacker per cell of 50 users, the attackers (one in each cell) may obtain between 11583 (64%) and 15874 (88%) time slots, depending on the scheme for assigning initial average throughput. When there are five attackers per cell, they can obtain between 14353 (79%) and 17136 (95%) time slots. Next, we compare the effect of the attack under this realistic situation where attackers do not know the CQIs of the victims to the the ideal situation discussed in Section 3.3. Figure 4(b) shows the fraction of time slots that the attackers obtain when they do not know the CQIs of the victim users compared to the case where they know the CQIs. In all cases, the attackers could still obtain more than 85% of the time slots that they would obtain in the ideal situation. In particular, if the PF scheduler uses user-provided initial average throughput, the attackers can obtain almost the same number of time slots as in the ideal situation.

As an illustration, we show the impact of the attack




(a) Time slots obtained by the attack in 30 seconds (18072 slots)




(b) Fraction of time slots obtained by attackers without knowing victims' CQIs compared to those with knowing victims' CQIs

**Figure 4. Performance of the attack without knowing victims' CQIs. Each sub-figure shows three curves, each representing a different scheme for assigning the initial average throughput.**

on VoIP communication, as cellular providers have recently started offering it. VoIP packets have a rigorous delay requirement: 0-0.15s delay is acceptable, 0.15s-0.40s delay might be tolerable, but longer delay is disruptive [20]. This delay budget is end-to-end, including the uplink delay from the sender ($U$), the transmission delay over the Internet ($T$, at least 0.1s across the continental USA), the downlink delay to the user ($D$), and other processing delay for VoIP ($O$, about 0.101s) [16]. One attacker can cause 0.81s downlink delay for victim users; therefore, 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 collude, the average downlink delay for a victim user increases to 1.80s, thus the end-to-end delay on users' VoIP applications is $0.10 + 1.80 + 0.10 = 2.01$s. Such excessive delay would make VoIP services useless. To illustrate, consider that geostationary satellite latency is between 240-280ms. Above illustrated attacks can elicit delays that are about 4-8 times longer.

# 4 Possible defense strategies

The above-illuminated attack exploits several vulnerabilities that combined enable malicious users to perform a denial-of-service attack on downlink cellular data service. To defend against these types of attacks, we outline some defense strategies that either eliminate relevant vulnerabilities or mitigate their impact. The following defense strategies could be implemented in current as well as future cellular systems.

## 4.1 Attack detection

There are three parameters that the base station can monitor in order to distinguish normal from under-attack operation; namely, *decrease of average user throughput*, *exorbitant number of handoffs* and *excessive retransmissions*. We can take advantage of these features in composing a defense strategy.

**Anomaly detection using average throughput** The base station can measure an average user's throughput during normal operation, either by simulation or by actual measurement. Then, it can compare the current throughput with recorded normal throughput. If their difference is above a certain threshold, this could indicate that the system is under attack. At this point, the base station can use several methods to mitigate the attack, including temporarily reverting to a scheduler that does not require user collaboration, such as round robin, while tracing the attack source.

**Number of handoffs per user** In a normal operation, users do not perform very frequent handoffs. On the other hand, attackers performs handoffs as frequently as one every 5 time slots, or one every 7.5-10ms. The base station can observe and record the number of handoffs performed per user over a period of time. If a user performs an unusually high number of handoffs in a given time, the base station can reject further handoff requests, thereby stopping the attack in that cell.

**Number of retransmissions per user** Due to mobility, it is normal for users to experience retransmissions. However, as attackers are overestimating their channel conditions while staying close to cell boundaries, the base station can detect an attack in progress if it observes that the number of continuous retransmissions per user (due to HARQ mechanism) is above a certain threshold value. False alarms may occur if the user is in an unfavorable condition, such as moving away from the BS or with a highly-variable channel condition.

## 4.2 Attack prevention

In this section, we first consider a set of variations of the PF scheduler and evaluate their effect on attack prevention. We then propose a new handoff scheme to be combined with a scheduler that can largely mitigate the effects of the attacks while considering network performance.

### 4.2.1 Variations of PF scheduler

We have discussed the PF scheduler so far. There are, however, various implementations of the PF scheduler, which we refered to as hybrid PF scheduler. While these modifications are proposed primarily for Quality of Service (QoS) purposes, we discuss their resilience against attacks from a security viewpoint.

**Priority queue** The base station can utilize priority queues to alleviate the impact of attacks outlined in the previous section. In particular, the base station can schedule traffic with delay constraints, such as VoIP traffic, with high priority, while other traffic, such as web browsing, can be scheduled with low priority. Because the number of high priority users is relatively small, these users have much better delay performance. Thus, the attack (in particular, attacks without handoff) effects of an attacker claiming to be high priority, will be mitigated. Its actual impact depends on the extent of system manipulation by the attacker. For instance, an attacker may want to opt out the priority set if he or she needs to stay dormant in order to lower his or her average throughput value. This can usually be achieved

by keeping the buffer at the base station empty or reporting extremely low CQI values. During the attack, the attacker can opt in the priority set through the following methods: masquerading as a high priority user, such as a VoIP user, triggering fast retransmissions, and having large queues (if the queue length is considered in scheduling decisions).

**Round-robin** Typically, system designers have to balance trade offs between short-term performance and overall throughput. To improve delay performance, (i.e., a form of short-term fairness), PF can be combined with round-robin scheduler with additional constraints such that each user should get scheduled for $m$ TTIs within a certain time window $w$, where $m <= w/N$ and $N$ is the number of users to be scheduled. Long-term fairness (e.g., in the pure PF scheduler), on the other hand, guarantees that each user obtains roughly the same amount of time slots over a long period of time (usually during the lifetime of a user, on the order of minutes). Choosing a lower $w$ and a larger $m$ improves short-term performance but at the expense of lowering the overall throughput. Conversly, enlarging $w$ and lowering $m$ improves overall throughput as the scheduler has more flexibility in choosing a user with good channel conditions but the expense of short-term performance.

#### 4.2.2 Robust handoff scheduler

All of the considered and proposed variations of the PF scheduler are confined within the realm of a single cell–none address the inter-cell issues, namely handoffs. Consider the case where a user moves from cell A to cell B and the two base stations could communicate and assign initial average values for the handoff user. The optimal initial value of the average throughput for the user in cell B may not necessarily be the average value in cell A. This new value will impact both security and system performance so it should be set high enough to deter attackers from attacking the system by initiating (frequent) handoffs but not cause excessive delays for normal users. Therefore, in terms of system performance, this value should be set to provide smooth transmission between cells so that the handoff user will not be any more or any less advantaged compared to the existing users in the cell. Additionally, to be fair, the newly assigned average value should reflect the transient behavior of the 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. Note that users can have different *average* channel conditions, e.g., depending on their distance to the base station. Relative channel fluctuation depends only on small-scale fading, such as scattering. Such 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.

When users experience statistically identical and independent relative channel fluctuations, multi-user diversity gain depends only on the number of users in a cell and the statistics of the channel fluctuation as shown in [8]. Assuming stationarity and ergodicity, the expectation of the average throughput of a user, $E(R)$[8], can be expressed as

$$E(R) = E(CQI)\frac{G(N)}{N} \tag{8}$$

where $CQI$ is a random variable, representing the user's channel condition, $N$ is the number of users in the cell, $E(CQI)/N$ is the average throughput of the user when $N$ users share the resource evenly without opportunistic scheduling, and $G(N)$ is the opportunistic scheduling gain, which is a function of $N$ and channel statistics. Opportunistic scheduling gain illustrates the performance gain of an opportunistic scheduling scheme over that of non-opportunistic 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, we have $G(N) \approx \log(N)$.

We propose the following heuristic to set the initial value of a 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 of after handoff, $R_B^{init}$, is set as

$$R_B^{init} = \frac{\sum_{i=1}^{N_A} R_A(i)}{E(CQI_A)\frac{G(N_A)}{N_A}} E(CQI_B)\frac{G(N_B)}{N_B} \cdot (1-\alpha)$$

$$R_B^{init} = \frac{\sum_{i=1}^{N_A} R_A(i)}{E(R_A)} E(R_B) \cdot (1-\alpha). \tag{9}$$

where $E(CQI_A)G(N_A)/N_A$ is the expected rate of the user in cell A (following Equation 8) and $E(CQI_B)G(N_B)/N_B$ is the expected rate of the user in cell B. In developing this formula one must be cautious not to set the initial value after handoff to be too high so as to disadvantage the user. Indeed, the value needs to

[8]We have dropped the time and user index from this equation.

be just high enough to deter attack. For example, setting the initial value naively to

$$R_B^{init} = \frac{R_A}{E(R_A)} E(R_B) \cdot (1 - \alpha). \quad (10)$$

may cause some legitimate users to experience unjustified delays. Typically, however, for a benign user, the ratio 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 $R_A \approx E(CQI_A)G(N_A)/N_A$. 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, Equation 9 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. In the presence of attackers, a malicious user may manipulate its value of $CQI_B$ for unfair advantages. We note that a user is often handoff to a cell with stronger signal strength, i.e., $E(CQI_B) \geq E(CQI_A)$. On the other hand, we do not expect $E(CQI_B)$ to be significantly higher than $E(CQI_A)$, otherwise, the handoff will be initiated earlier. Therefore, to deter attackers and to avoid estimations of $CQI_A$ and $CQI_B$, we can set

$$R_B^{init} \approx \frac{\sum_{i=1}^{N_A} R_A(i)}{\frac{G(N_A)}{N_A}} \frac{G(N_B)}{N_B} \cdot (1 - \alpha). \quad (11)$$

Our initial simulations support that this handoff scheme can deter attackers from resetting their throughput values to prolong attacks. However, in our future work, we will further evaluate its performance and extend the handoff study to general opportunistic schedulers.

## 5 Future Work

Cellular network considers that all mobile phones are part of its Trusted Computing Base (TSB). This assumption enables attackers to exploit its trust and perform DoS attacks against the network as demonstrated in our attack simulations. In lieu of their effectiveness, we have augmented the handoff mechanism in order to make the scheduler more robust. Initial results have been very promising. Nevertheless, the proposed scheme is in a preliminary stage. There are important issues to be addressed. The scheme applies to the case where users have statistically identical rate fluctuation, which may not always be true in practice. In addition, the estimation of $G(\cdot)$ can be difficult without the precise knowledge of channel statistics. Therefore, an important and practical issue is to determine the handoff initial value given only the current average rate of users in both cells. More importantly, we plan to extend the handoff study to general opportunistic schedulers. We also plan to evaluate the handoff study in the presence of a large number of attackers (due to device viruses or malware).

## 6 Related work

Studies on the security of 3G networks began to appear in recent years [9, 26, 33]. Sridharan et al. modeled the uplink channel from mobile devices to the base station in EV-DO and suggested that malicious users could modify their power transmission levels to cause interference on honest users [36]. By contrast, our work concentrates on the downlink given that downlink bandwidth in 3G networks is considerably higher than uplink bandwidth. Furthermore, we present not only threats but also attacks that exploit these threats.

Denial of service (DoS) attacks on cellular networks have attracted a lot of attention as resources on cellular networks are much more limited than those on the Internet. Agarwal et al. [2] 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. [13] presented a denial-of-service attack by sending a sufficient number of SMS messages per second to a range of cellular phones in the same area. An attacker would need only a single computer with a broadband network access to disrupt a network in a major city by saturating control channels shared between voice calls and SMSs. Traynor et al. [40] evaluated this attack using a highly accurate GSM simulator and proposed mitigation strategies. [28] warns that the paging channel is another scarce resource that an attacker on the Internet can overwhelm to cause a DoS attack. Furthermore, Traynor et al. pointed out that in spite of numerous efforts to securely overlay a packet-switched network onto a circuit-switched network, mechanisms responsible for connection establishment are still vulnerable to low-bandwidth DoS attacks [41]. Racic et al. [32] 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. Finally, jammers [45] can disrupt cellular networks as well. All these studies focused on attacks originating from outside the cellular network, usually from the Internet. Particularly, jammers are often ineffective in causing a DoS on the cellular infrastructure on a large scale since each jammer covers a very limited area. They can also be very difficult to obtain and easy to detect [15]. In contrast, our work focuses on DoS attacks from inside the cellular network and uses existing mobile devices, such as cellular phones and 3G

cards. Compared to jamming attacks, our attack is much more difficult to detect because it follows wireless media access protocols.

Significant amount of research has been conducted on efficient resource sharing in cellular networks. In particular, opportunistic scheduling algorithms have been studied extensively [23, 27, 42]. However, prior work focused on improving system performance under various system constraints and requirements, including the effect on TCP performance [5, 12], instability [3], and multi-cell scheduling [10]. In contrast to these studies, we consider the threat of malicious users and their impact on the PF scheduler. While (artificial) handoff has been considered for load-balancing purpose in [10, 35], no one has studied how to assign good initial values for handoff users, to the best of our knowledge.

Recently, concurrently with and independently of our work, Bali et al. have showed that a long lived network flow of a victim mobile device can be starved by a sudden arrival of packets to another offending mobile device whose buffer had been empty for a period of time [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.

In a simulation comparing our attack to theirs under the same network condition, our attack occupied more than twice as many consecutive slots in a cell with only 2 users, and more than three times as many consecutive slots in a cell with 50 users.[9] Moreover, since their attack exploits the fact that a user's average rate drops when the user's buffer is empty, their attack can be mitigated by limiting the decrease of average throughput when the buffer is empty [46].

Finally, in addition to the vulnerability of fake CQI reports, we also discovered a vulnerability in the handoff procedure that can be exploited in combination with fake CQIs to attack the PF scheduler. Our attack, exploiting multiple attack vectors, significantly magnifies the effect of the attack and causes severe DoS to the cellular network. Furthermore, we proposed a robust handoff algorithm that manages to mitigate the attack.

## 7 Conclusion

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

- The network trusts mobile devices to report truthful CQIs, which the PF scheduler uses without verification for assigning time slots. Therefore, 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.

We have studied a series of attacks on the PF scheduler by exploiting the above vulnerabilities. Our simulations show that just one attacker per cell can decrease the throughput and increase the delay of victim users significantly, and can disrupt time-sensitive data services, such as VoIP. Moreover, multiple attackers in the same cell can collaborate to aggravate the attack. To defend against the attacks, we first propose a set of attack detection schemes. Then, we discuss a variety of modifications of the PF scheduler and their resilience to the attacks. Finally, we propose a handoff heuristic that significantly mitigates the impact of the aforementioned attacks.

[9] In a cell with only two users, our attack occupied 1198 consecutive slots while Bali et al.'s method occupied only 529 slots. In a cell with 50 users, our method occupied 65 slots while theirs occupied only 20 slots.

## References


[1] 3GPP. 3g wlan - trust model. `http://www.3gpp.org/ftp/tsg_sa/WG3_Security/TSGS3_25_Munich/Docs/PDF/S3-020523.pdf`.

[2] N. Agarwal, L. Chandran-Wadia, and V. Apte. Capacity analysis of the GSM short message service. In *National Conference on Communications*, 2004.

[3] M. Andrews. Instability of the proportional fair scheduling algorithm. In *IEEE Transactions on Wireless Communications*, 2004.

[4] S. Z. Asif. Aligning business and technology strategies–an evolution of a third generation wireless technology. In *Engineering Management Conference*, 2002.

[5] M. Assaad, B. Jouaber, and D. Zeghlache. Effect of TCP on UMTS-HSDPA system performance and capacity. 2004.

[6] S. Bali, S. Machiraju, H. Zang, and V. Frost. On the performance implications of proportional fairness (pf) in 3g wireless networks. In *In Proceedings of Passive and Active Measurements Conference*, 2007.

[7] 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. In *IEEE Communications Magazine*, July 2000.

[8] S. Borst. User-level performance of channel-aware scheduling algorithms in wireless data networks. In *Proceedings of IEEE INFOCOM*, 2003.

[9] A. Bovosa. Attacks and counter measures in 2.5G and 3G cellular IP networks. In *Juniper White Paper*, 2004.

[10] T. Bu, L. Li, and R. Ramjee. Generalized proportional fair scheduling in third generation wireless data networks. In *INFOCOMM*, 2006.

[11] E. F. Chaponniere, P. Black, J. M. Holtzman, and D. Tse. Transmitter directed multiple receiver system using path diversity to equitably maximize throughput. U.S. Patent No. 6449490, 2002.

[12] J.-H. Choi, J.-G. Choi, and C. Yoo. Analyzing the impact of proportional fair scheduler on TCP performance. 2005.

[13] W. Enck, P. Traynor, P. McDaniel, and T. L. Porta. Exploiting open functionality in SMS-capable cellular networks. In *12th ACM Conference on Computer and Communications Security (CCS'05)*, Nov. 7-11, 2005.

[14] Ericsson. WCDMA evolved - the first step – HSDPA. `http://www.ericsson.com/technology/whitepapers/wcdma_evolved.pdf`.

[15] FCC. Communications act of 1934: as ammended by telecom act of 1996. `http://www.fcc.gov/Reports/1934new.pdf`.

[16] B. Goode. Voice over internet protocol (VoIP). In *IEEE*, 2002.

[17] M. Grosslauser and D. Tse. Mobility increases the capacity of wireless ad hoc networks. In *IEEE Infocom*, April 2001.

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

[19] 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`.

[20] ITU-T. One-way transmission time. ITU-T Recommendation G.114, 1996.

[21] A. Jalali, R. Padovani, and R. Pankaj. Data throughput of CDMA-HDR a high efficiency-high data rate personal communication wireless system. In *Proceedings of IEEE Vehicular Technology Conference 2000-Spring*, volume 3, 2000.

[22] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[23] R. Knopp and P. Humblet. Information capacity and power control in single-cell multiuser communications. In *Proceedings of the ICC*, 1995.

[24] T. E. Kolding. Link and system performance aspects of porportional fair scheduling in WCDMA/HSDPA. In *Vehicular Technology Conference*, 2003.

[25] T. E. Kolding. Link and system performance aspects of proportional fair scheduling in wcdma/hsdpa. In *VTC*, 2003.

[26] K. Kotapati, P. Liu, Y. Sun, and T. F. L. Porta. A taxonomy of cyber attacks on 3G networks. In *Technical Report NAS-TR-0021-2005, Network and Security Research Center, Department of Computer Science and Engineering, Penn State University*, 2005.

[27] X. Liu, E. K. P. Chong, and N. B. Shroff. A framework for opportunistic scheduling in wireless networks. *Computer Networks*, 41(4):451–474, March 2003.

[28] P. Mutaf and C. Castelluccia. Insecurity of the paging channel in the wireless internet: A denial-of-service attack that exploits dormant mobile IP hosts. In *3rd Workshop on Applications and Services in Wireless Networks*, 2003.

[29] S. Nanda, K. Balachandran, and S. Kumar. Adaptation techniques in wireless packet data services. *IEEE Communications Magazine*, 38(1):54–64, January 2000.

[30] Novatel. Novatel merlin u870 pc card. `http://www.novatelwireless.com/products/merlin/merlin-u870.html`.

[31] Qualcomm. Hsdpa for improved downlink data transfer. October 2004.

[32] R. Racic, D. Ma, and H. Chen. Exploiting MMS vulnerabilities to stealthily exhaust mobile phones' battery. In *IEEE SecureComm*, 2006.

[33] F. Ricciato. Unwanted traffic in 3G networks. In *ACM SIGCOMM Computer Communication Review, Volume 36, Issue 2*, 2006.

[34] P. Rysavy. Data capabilities: GPRS to HSDPA and BEYOND. `http://www.cingular.com/b2b/content/downloads/DataCapabilities_beyond.pdf`.

[35] 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. In *MobiCom '04: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 302–314, New York, NY, USA, 2004. ACM Press.

[36] A. Sridharan, R. Subbaraman, and R. Guerin. Uplink scheduling in the EV-DO rev. a system: An initial investigation. In *Sprint ATL Research Report Nr. RR06-ATL-080139*, 2006.

[37] H. Suzuki. Statistical model for urban radio propagation. In *IEEE Transactions on Communications*, July 1977.

[38] A. Systems. Hsdpa mobile broadband data. 2005.

[39] Telefono. Homebrew mobile phone club. `http://telefono.revejo.org/`.

[40] P. Traynor, W. Enck, P. McDaniel, and T. L. Porta. Mitigating attacks on open functionality in SMS-capable cellular networkss. In *12th Annual International Conference on Mobile Computing and Networking MOBICOMM*, 2006.

[41] P. Traynor, P. McDaniel, and T. L. Porta. On attack causality in internet-connected cellular networks. In *USENIX Security Symposium (SECURITY)*, 2007.

[42] D. Tse and P. Viswanath. *Fundamentals of Wireless Communication*. Cambridge, 1 edition, 2005.

[43] K. Tuna. Hacking EVDO. In *Defcon 15*, 2007.

[44] V. Vanghi, A. Damnjanovic, and B. Vojcic. *The cdma2000 System for Mobile Communications*. Prentice Hall, 2004.

[45] Wikipedia. Cell phone jammer. `http://en.wikipedia.org/wiki/Cell_phone_jammer`.

[46] J. Yang, Z. Yifan, W. Ying, and Z. Ping. Average rate update mechanism in proportional fair scheduler in hdr. In *Globecom*, 2004.