5

Vulnerabilities in the Short Messaging Service (SMS)

For large portions of the population, the *Short Messaging Service* (SMS) has eclipsed voice telephony as the dominant means of communication. Noted for its discreet nature, SMS allows mobile subscribers to interact via concise, text-only messages. During its relatively short lifetime, SMS has transformed from a niche service to a system with a larger user-base than the Internet. While the benefits of this service have been tremendously positive from the perspectives of both user satisfaction and provider revenue, the inclusion of SMS in cellular networks creates significant new security issues.

In this chapter, we examine vulnerabilities created by the introduction of text messaging. Our discussion begins with an evaluation of the evolution of SMS. We then present an in-depth description of the wireless portion of the network and then discuss how competition for channels shared by both voice and text messages allows low bandwidth attacks to deny voice service to major metropolitan areas. Specifically, we apply modeling and simulation to demonstrate that an adversary with the bandwidth available to a cable modem is capable of denying service to more than 70% of calls and text messages for Manhattan. For networks in which a 1% blocking rate is unacceptable, such a degredation represents a serious operational crisis. Our investigation then turns to current countermeasures and shows why the use of "edge solutions" is not a sufficient means of preventing such attacks. We then suggest, characterize and measure the effectiveness of more appropriate mechanisms from the areas of queue management and resource provisioning.

More importantly, this chapter provides one of the first perspectives on the impact of connecting the telecommunications infrastructure and the Internet. Largely secured by its physical isolation from other systems in the past, national and international critical infrastructure are now potentially threatened by new and unpredictable attack vectors.

5.1 History and Description

The idea of a text communication mechanism as part of a cellular network was discussed widely in the early 1980's. The uses of such a service, however, varied widely. While the majority of the community imagined SMS as a mechanism to alert users of events such as voice mail and network outages, others advocated a system capable of supporting various data collection devices (e.g., telemetry). To meet the wishes all parties, the original SMS standard document (1985) [123] detailed three general functions: Short Message Mobile Terminated (from the network to the device), Short Message Mobile Originated (from a device to the network), and Short Message Cell Broadcast (from the network to all devices in an area).

Uptake of the service was slow – the first commercial text message was transmitted some seven years later in 1992 [79]. Customer use remained flat throughout until nearly the end of the decade; however, the introduction of inter-provider messaging agreements and pre-paid user plans significantly boosted the popularity of the service. By the year 2000, 5 billion messages were being sent per month. In 2005, that number increased by nearly two orders of magnitude with an estimated 1 trillion messages sent worldwide [79].

As it exists today, SMS is a text-only service that delivers message containing up to 160 characters. By default, messages are encoded in an alphabet supporting an extended Latin character set known as the GSM 7-bit default alphabet. Non-Latin character sets, including Arabic, Chinese and Russian, can be supported by an alternate 16-bit encoding. The transmission of longer messages is typically supported by splitting text into multiple messages. Complimentary services, such as MMS, can be used to support messages containing images, audio and formatting.

5.2 Delivering Messages

Before discussing vulnerabilities introduced by text messaging, we provide an overview of message delivery in a GSM network. Readers that do not possess a strong background in cellular networks are encouraged to reference Chapter 3 for additional architectural information.

5.2.1 Submitting a Message

Text messages can be sent to mobile phones via one of two general methods – from another phone or mobile device as a *mobile originated* (MO) message or through a variety of *External Short Messaging Entities* (ESMEs). ESMEs encompass devices and interfaces ranging from email and provider-run web-based messaging portals to voice mail, paging and bulk advertising systems. Whether a message is sent from another device in the network or an ESME, all messages are first processed by a *Short Messaging Service Center* (SMSC).

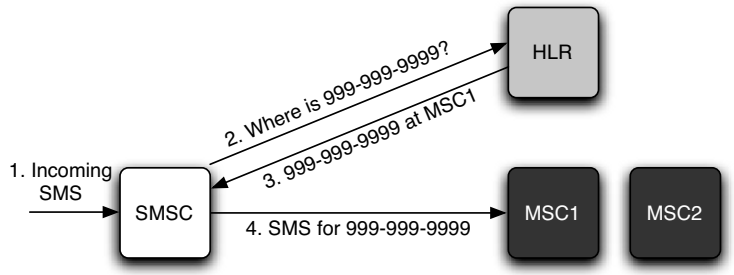

Fig. 5.1. A high-level description of text message routing. Messages 1) arrive at an SMSC from ESMEs and devices within the network. The SMSC then 2) queries the HLR to find the location of the targeted device. The HLR responds 3) with the address of the MSC serving the phone. The SMSC then 4) places the formatted message into a MAP packet and forwards it to the appropriate MSC.

These servers are responsible for the "store-and-forward" protocol that eventually delivers text messages to their intended destination. Accordingly, service providers supporting text messaging must have at least one SMSC in their network. Due to the rising popularity of this service, however, it is becoming increasingly common for service providers to support multiple SMSCs in order to increase capacity.

5.2.2 Routing a Message

When a text message arrives at an SMSC, the contents of incoming packets are examined and, if necessary, converted and copied into SMS message format. First, messages are converted into the 7-bit default alphabet, which includes all characters in the ASCII character set and a number of other accented letters [25]. The SMSC then attempts to determine the location of the targeted phone by querying the corresponding HLR with a *SendRoutingInfoForShortMsg* request. If the targeted mobile phone is available (i.e., powered on), the HLR will respond with the address of the serving MSC. Otherwise, messages destined for unreachable are kept in the SMSC until the device becomes available. Assuming that the address of the serving MSC is returned, a MAP packet is then created and tagged with the *SMSDeliveryForward* (SMDFWD) operation specifier and forwarded. Figure 5.1 provides a high-level overview of this procedure.

When a text message arrives, the MSC begins the process of locating the targeted device. To do this, the MSC queries its associated VLR, which stores local copies of device service profiles. If the targeted device is currently in use (i.e., in a phone call), the base station nearest to the device is known and the messages is forwarded. If instead the targeted device is on but not in use, the MSC initiates the paging process described below.

5.2.3 Wireless Delivery

An area of wireless coverage is called a *cell*. Cells, which are serviced by *base stations*, are typically partitioned into multiple (usually three) smaller regions called sectors. Users standing on opposite sides of a base station are therefore likely to be operating using different resources.

The air interface, or wireless portion of the network, is divided into two general classes of logical channels – the *Control Channels* (CCHs) and *Traffic Channels* (TCHs). Traffic channels are responsible for supporting voice telephony once calls have been established. CCHs are therefore responsible for implementing all other operations necessary to run the network. Accordingly, the CCHs are partitioned into two additional classes of channels – the Common CCH and Dedicated CCHs. The Common CCH, which includes the *Broadcast Control Channel* (BCCH) is responsible for operations ranging from alerting devices of incoming calls or messages (called paging) to helping devices synchronize with the network. Accordingly, all mobile devices not currently engaged in voice or data calls periodically listen to the Common CCH. Dedicated CCHs provide information related to call setup for individual phones. As detailed below, call and SMS delivery requires the use of both classes of CCH.

When a device is not actively communicating with the network, the MSC does not keep track of the base station closest to that phone. Accordingly, upon arrival of a new call or text message, the MSC must search multiple connected base stations to locate the device. The traditional strategy to finding a phone quickly is to flood all attached base stations with paging requests. As expected, flooding requires significant resources in terms of processing power and bandwidth at the MSC. While a number of solutions reducing resource use have been proposed [27, 112], the traditional flooding mechanism is typically used in current networks because its latency and implementation complexity are low.

Each of the base stations receiving a paging message from the MSC transmits the incoming request on the *Paging Channel* (PCH). Instead of broadcasting the phone number of the targeted device, however, the network transmits a *Temporary Mobile Subscriber Identifier* (TMSI). The use of a TMSI, which is typically updated after each interaction with the network, makes tracking specific phones by wireless eavesdroppers difficult. When a device hears its TMSI broadcast on the PCH, it attempts to alert the network of its presence and availability by means of the *Random Access Channel* (RACH). The performance of this channel is discussed in greater depth in Chapter 6. Upon receiving a response, the MSC drops outstanding paging messages in other cells and the base station authenticates the phone. The base station points the phone to listen to a specific dedicated control channel via the *Access Grant Channel* (AGCH). These dedicated control channels, known as the *Standalone Dedicated Control Channels* (SDCCHs), allow the network to perform authentication via the triplet tokens described in Chapter 3, assign

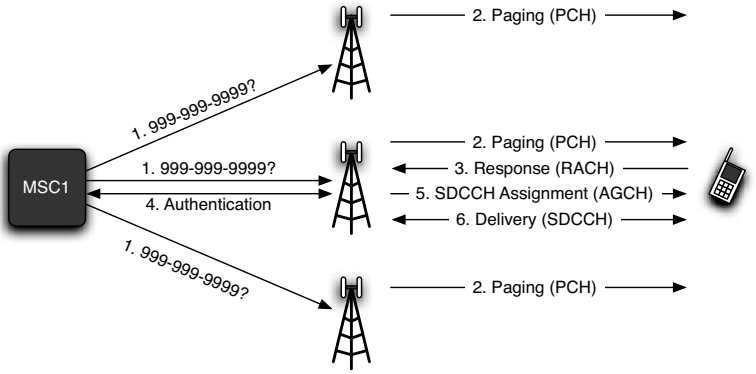

Fig. 5.2. A description of the wireless delivery process for SMS. After receiving a text message, 1) the MSC sends paging messages to multiple base stations to find the targeted phone. The base stations then 2) transmit the paging request on the PCH. When a phone hears its TMSI, 3) it responds to the base station on the RACH, which in turn 4) alerts the MSC that the phone is available and needs authentication. The base station then 5) assigns an SDCCH to the phone via the AGCH. The phone 6) then authenticates with the base station and receives the text message.

a new TMSI and deliver the contents of a text message [19]. If the network was delivering an incoming voice call instead, the delivery of an SMS message would be replaced by a pointer to a TCH upon which the call could be conducted.

We summarize these steps for simplicity in Figure 5.2.

5.3 Identifying System Bottlenecks

The closed nature of telecommunications networks makes testing specific components difficult. In spite of this, much can still be learned about these systems simply by interacting with them. In this section, we apply a technique known as gray-box testing [34, 46] to learn about the internal behavior of operational cellular networks not specified in standards documents. We characterize queuing policies, delivery rates and interfaces and demonstrate that the imbalance between resource injection and delivery rates creates the necessary precondition for a DoS attack.

Very small volumes of traffic were injected during off-peak periods in order to observe network behavior; the attack detailed in Section 5.5 was not launched against any network. Experiments damaging or disrupting such networks without the specific permission of the provider and/or government approval are illegal and should not be conducted.

5.3.1 Queue Management

Queuing determines the behavior of flows through a network. As shown in the previous subsection, two points in the network are of particular interest for such analysis – the SMSC and the targeted device. We therefore characterize the behavior of both of these elements in a number of networks. While our attack is focused on GSM networks, we explore SMSCs and devices attached to multiple types of cellular networks in order to develop a broader appreciation for the diversity of policy across systems.

Queuing in the SMSC

SMSCs are the core of text messaging operations in a cellular network. While every text message sent and received in the system passes through them, the capacity of an SMSC is limited by a number of practical considerations. Most critically, the "store-and-forward" nature of message delivery is governed by the storage capacity of the SMSC. We therefore characterize the buffering capacity and queue eviction policies to show their impact on large volumes of text messages.

The queuing and eviction policies for SMSCs were assessed by periodically sending text messages to powered-off phone. We then sent a few hundred text messages containing sequential identifiers at a rate of once per minute. The device was then powered-on and reconnected to its default network, at which time as many messages as possible were delivered. The networks of US providers AT&T[1], Verizon and Sprint were all tested using the above methodology.

Our experiments found a significant range of queuing and eviction policies. AT&T's SMSC buffered a maximum of 400 messages and delivered them in order of submission. In contrast, Verizon stored only 100 messages. More interestingly, however, Verizon discarded the first few hundred messages and delivered only the final 100. Sprint also differed from the other providers by storing only 30 messages. Like AT&T, messages sent after the user's SMSC buffer was filled experienced tail drop loss in the Sprint network.

Queuing in Phones

Mobile phones are highly constrained computing devices. Their processing ability, memory and access to power all pale in comparison to modern general purpose computing platforms. Understanding this, cellular networks allow phones to alert the network when their buffering capacity has been met. In GSM, phones send a MAP message containing the *Mobile-Station-Memory-Capacity-Exceeded-Flag* (MCEF) to the HLR [19]. When this flag is set in a

[1] At the time of testing, AT&T Wireless was in the initial phases of becoming Cingular Wireless. At the time of this writing, Cingular Wireless has once again become AT&T Wireless.

Table 5.1. Mobile Phone SMS Capacity

Device	Capacity (messages)
Nokia 3560	30
LG 4400	50
Treo 650	500*

* 500 messages depleted a full battery.

user's profile at the HLR, responses to *SendRoutingInfoForShortMsg* requests from the SMSC indicate that the phone is unavailable.

Determining the message storage capacity of all available phones is not practical. Instead, we selected phones exemplifying the range of capabilities available to such devices at the time of our original testing. To represent low, medium and high-end devices, we measured the capacity of a Nokia 3560 (AT&T), and LG 4400 (Verizon) and a Treo 650 with removable 1GB memory stick (Sprint), respectively. Capacity was tested by clearing all messages from the targeted device and slowly sending messages (one per minute) until the inbox was filled. As shown in Table 5.1, the capacity of both the low and medium-end phones were quickly reached. The experiments on the high-end phone resulted in a capacity of 500 messages; however, this limit is the result of the battery being drained and not the exhaustion of storage.

Implications of Queue Management

The above testing provides a number of insights into crafting targeted SMS attacks against the telecommunications infrastructure. Most importantly, the small buffers in both SMSCs and end devices suggest that large scale attacks must be distributed across multiple end devices in order to avoid overfilling SMSC and device buffers. This information also suggests that attacks targeted at individuals are also possible. Because we have shown that it is possible to cause a network to lose messages, an adversary can fill a user's SMSC queue as a means of preventing them from receiving a specific message. We investigate the implications of these observations in Section 5.5.

5.3.2 Message Injection

Key to any DoS attack is the rate at which an adversary can inject messages. In this section, we examine a sampling of the many interfaces through which targeted text messaging attacks can be launched. We use publicly available specifications to quantify conservative values for message insertion rates. Finally, we show that the imbalance between message injection and delivery time allows an adversary to use low-bandwidth attacks to impact network services.

The number of methods by which malicious text messages can be injected into a network is staggering. For instance, compromised mobile devices or

those running programs with malicious side effects can create significant volumes of text messages. Given that 2.7% of cellular users download games each month [97] and a growing set of Blutoooth-based exploits [187, 62, 65], the presence of such malicious software is increasingly possible. As mentioned previously, ESMEs including service provider web interfaces, email, and instant messaging clients would open additional portals to infected PCs throughout the Internet. Even larger pipes running the *Short Messaging Peer Protocol* (SMPP) connect bulk SMS senders to the telecommunications infrastructure. With advertised rates of 30-35 messages text messages per second per client and additional services offering ten times that bandwidth [169], an adversary could hire such providers to inject significant additional messages to those sent through the free interfaces above. Given the preponderance of methods to inject text messages and individual SMSC capacities above 20,000 messages per second [29], we conservatively estimate that between several hundred and several thousand messages per second can be submitted to a network.

When the injection rate of messages exceeds the network's ability to deliver them, DoS attacks become possible. Given that SMSCs can process over 20,000 messages per second and our ability to inject several thousand messages injected per second, a superficial examination would suggest that such attacks are not possible. However, not all portions of the network are as well provisioned as the SMSCs. To illustrate this, we perform additional gray box testing and measure message injection and delivery times. For the former, we use PERL scripts that interface with the service provider web interfaces to send messages to targeted phones. An average message submission time of 0.71 seconds was recorded across all three providers. Precise measurement of delivery time were more difficult because of experimental setup (clock synchronization between the sending desktop and a phone is not a simple task). Informally, we observed an average of 7-8 seconds. These experiments demonstrate approximately an order of magnitude difference between the time required to insert a message and the time needed to delivery it.

In combination with standards information discussed in Section 5.5, the experiments conducted in this section indicate that the necessary precondition for DoS on cellular networks, an imbalance between message injection and delivery rates, is present in operational systems.

5.4 Efficient Device Targeting

The ability to successful attack on a mobile phone network requires the adversary to do more than simply attempt to send text messages to every possible phone number. Much like the creation of hit-lists for accelerated worm propagation across the Internet [168], it is possible to efficiently create a database of potential targets within a cellular phone network. The techniques below, listed from the most coarse to fine-grain methods, represent a subset of techniques

for creating targeted attacks. The combination of these and other methods can be used to create extremely accurate hit-lists.

The most obvious first step would be simply to attempt to capture phone numbers overheard on the air interface. Because of the use of TMSIs over the air interface, this approach is not possible[2]. We therefore look to the web as our source of data.

5.4.1 NPA/NXX

The United States, Canada, and 18 other nations throughout the Caribbean adhere to the *North American Numbering Plan* (NANP) for telephone number formatting. NANP phone numbers consist of ten digits, which are traditionally represented as "NPA-NXX-XXXX[3]". These digit groupings represent the area code or *Numbering Plan Area*, exchange code[4], and terminal number, respectively. Traditionally, all of the terminal numbers for a given NPA/NXX prefix are administered by a single service provider.

A quick search of the Internet yields a number of websites with access to the NPA/NXX database. Responses to queries include the name of the service provider administering that NPA/NXX domain, the city where that domain is located and the subdivision of NPA/NXX domains among a number of providers. For example, in the greater State College, PA region, 814-876-XXXX is owned by AT&T Wireless; 814-404-XXXX is managed by Verizon Wireless; 814-769-XXXX is supervised by Sprint PCS.

This information is useful to an attacker as it reduces the size of the domain to strictly numbers administered by wireless providers within a given region; however, this data does not give specific information in regards to which of the terminals within the NPA/NXX have been activated. Furthermore, as of November 23, 2004, this method does not account for numbers within a specific NPA/NXX domain that have been transferred to another carrier under new number portability laws. Nonetheless, this approach is extremely powerful when used in conjunction with other methods, as it reduces the amount of address space needed to be probed.

5.4.2 Web Scraping

As observed in the Internet, a large number of messages sent to so-called "dark address space" is a strong indicator that an attack is in progress. A more refined use of domain data, however, is readily available.

---

[2] It may be possible to match TMSIs to a specific user if the adversary can overhear an initial assignment. During this exchange, the IMSI is transmitted without encryption.

[3] Numbers in the last two subsets can take the form of N(2-9) or X(0-9)

[4] The "NXX" portion of a phone number is sometimes referred to as the "NPX" or *Numbering Plan Exchange*.

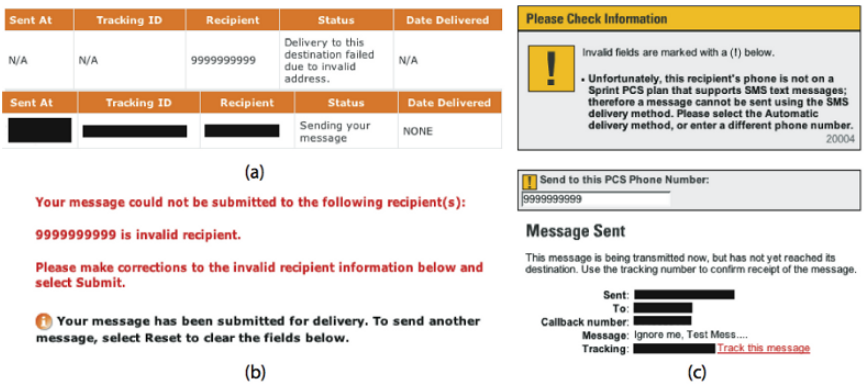

Fig. 5.3. The negative (top) and positive (bottom) response messages created by message submission to a) Verizon, b) Cingular and c) Sprint PCS. Black rectangles have been added to preserve sensitive data.

Web Scraping is a technique commonly used by spammers to collect information on potential targets. Through the use of search engines and scripting tools, these individuals are able to gather email addresses posted on web pages in an efficient, automated fashion. These same search tools can easily be harnessed to collect mobile phone numbers listed across the web. For example, the query `Cell 999-999-0000..9999` at Google yields a large number of hits for the entire range of the NPA/NXX "999-999-XXXX". Through our own proof-of concept scripts, we were able to collect 865 unique numbers from the greater State College, PA region, 7,308 from New York City and 6,184 from Washington D.C. with minimal time and effort.

The difficulty with this method, much like the first, is that it does not give a definitive listing of numbers that are active and those that are not. As personal web pages are frequently neglected, the available information is not necessarily up to date. Accordingly, some portion of these numbers could have long since been returned to the pool of dark addresses. Furthermore, due to number porting, there is no guarantee that these numbers are still assigned to the service provider originally administering that domain. Regardless, this approach significantly narrows down the search space of potential targets.

5.4.3 Testing Phone "Liveness"

All of the major providers of wireless service in the United States offer a website interface through which anyone can, at no charge to the sender, submit SMS messages. If a message created through this interface is addressed to a subscriber of this particular provider, the message is sent to the targeted mobile device and a positive acknowledgment is delivered to the sender. A message is rejected from the system and the user, depending on the provider, is returned an error message if the targeted device is a subscriber of a different

provider or is addressed to a user that has opted to turn off text messaging services. An example of the both the positive and negative acknowledgments is available in Figure 5.3. Of the service providers tested (AT&T Wireless, Cingular, Nextel, Sprint PCS, T-Mobile and Verizon Wireless[5]), only AT&T did not respond with a positive or negative acknowledgment; however, it should be noted that subscribers of AT&T Wireless are slowly being transitioned over to Cingular due to its recent acquisition.

The positive and negative acknowledgments can be used to create an extremely accurate hit-list for a given NPA/NXX domain. Every positive response generated by the system identifies a potential future target. Negative responses can be interpreted in multiple ways. For example, if the number corresponding to a negative response was found through web scraping, it may instead be tried again at another provider's website. If further searching demonstrates a number as being unassigned, it can be removed from the list of potential future targets.

While an automated, high speed version of this method of hit-list creation may be noticed for repeated access to dark address space, an infrequent querying of these interfaces over a long period of time (i.e., a "low and slow" attack) would be virtually undetectable.

A parallel result could instead be accomplished by means of an automated dialing system; however, the simplicity of code writing and the ability to match a phone to a specific provider makes a web-interface the optimal candidate for building hit-lists in this fashion.

5.4.4 Additional Collection Methods

A number of specific techniques can also be applied to hit-list development. For example, a worm could be designed to collect stored phone numbers from victim devices by address book scraping. In order to increase the likelihood that a list contained only valid numbers, the worm could instead be programmed to take only the numbers from the "Recently Called" list. The effectiveness of his method would be limited to mobile devices running specific operating systems. The interaction between many mobile devices and desktop computers could also be exploited. An Internet worm designed to scrape the contents of a synchronized address book and then post that data to a public location such as a chat room would yield similar data. Lastly, Bluetooth enabled devices have become notorious for leaking information. Hidden in a busy area such as a bus, subway or train terminal, a device designed to collect this sort of information [187, 180] through continuous polling of Bluetooth-enabled mobile phones in the vicinity would quickly be able to create a large and temporally

[5] Since the original experiments, the above providers have changed. AT&T Wireless was absorbed by Cingular, which recently changed its name back to AT&T. Sprint and Nextel have also merged into a single entity.

accurate hit-list. If this system was left to run for a number of days, a correlation could be drawn between a phone number and a location given a time and day of the week.

5.5 Modeling Denial of Service

Given the existing bottlenecks and the ability to create hit-lists, we now discuss attacks against cellular networks. An adversary can mount an attack by simultaneously sending messages through the numerous available portals into the SMS network. The resulting aggregate load saturates the control channels thereby blocking legitimate voice and SMS communication. Depending on the size of the attack, the use of these services can be denied for targets ranging in size from major metropolitan areas to entire continents.

5.5.1 Attacking Individuals

In 2002, anonymous individuals inundated spammer Alan Ralsky with thousands of mail-order catalogs on a daily basis. Through the use of simple scripting tools [48], these individuals subscribed Ralsky to postal mailing lists at a much faster rate than he could possibly be removed. In so doing, Ralsky's ability to receive normal postal mail at his primary residence was all but destroyed.

Similar attacks on text messages are also possible. Given the SMSC queue management policies discovered in Section 5.3.1, an adversary can prevent a victim from receiving useful messages. For example, a jealous ex-lover may wish to keep a message from being delivered; a stock trader may want to delay updates received by competitors; an attacker may want to keep a systems administrator from receiving a notification.

This attack is accomplished by flooding the user with messages. This results in one of three outcomes: a buffer somewhere overfills and the message is lost, the message is delayed longer than its shelf-life[6], or the user does not notice the message due to the deluge of meaningless messages.

In many cases, an attack allowing intentional message loss is ideal for the adversary. Mobile phones, like other embedded devices, have significant memory constraints, thereby limiting the number of messages a phone can hold. Once the phone can no longer receive messages, the service provider's network begins to buffers all subsequent messages. For reasons of practicality, providers impose limitations on the number of messages the network can store per user. Thus, if the adversary can exceed this value, messages become lost.

Message loss can occur throughout the network. As observed with the Nokia 3560, when the buffer became full, any previously read messages were automatically deleted. While this occurrence was isolated to the firmware of

[6] An SMS weather notification is useless if you are already stuck in the rain.

a specific phone, the potential to remotely maliciously destroy a user's data exists.

The onslaught of large numbers of packets helps accomplish the remaining two attack outcomes. During our gray-box testing in Section 5.3.1, the delivery of all 400 packets stored in the SMSC took almost 90 minutes even with the constant monitoring and clearing of phone buffers. Temporally critical messages were potentially delayed beyond their period of usefulness. Additionally, the use of the "Clear Inbox" function significantly increases the possibility of a user accidentally deleting a legitimate text message that arrived among the attack messages.

While deleting an immense number of text messages is taxing on the user, the receipt of large amounts of data consumes significant battery power. This leads to yet another targeted DoS attack, a battery depletion attack. After the publication of our original work, Racic et al. [142] discussed a similar battery depletion attack using the data network.

5.5.2 Metropolitan Area Service

As discussed in Section 5.2, the wireless portion of SMS delivery begins when the targeted device hears its *Temporary Mobile Subscriber ID* (TMSI) over the *Paging Channel* (PCH). The phone acknowledges the request via the *Random Access Channel* (RACH) and then proceeds with authentication and content delivery over a *Standalone Dedicated Control Channel* (SDCCH).

Voice call establishment is very similar to SMS delivery, except a *Traffic Channel* (TCH) is allocated for voice traffic at the completion of control signaling. The advantage of this approach is that SMS and voice traffic do not compete for TCHs, which are held for significantly longer periods of time. Therefore, TCH use can be optimized such that the maximum number of concurrent calls is provided. Because both voice and SMS traffic use the same channels for session establishment, contention for these limited resources still occurs. Given enough SMS messages, the channels needed for session establishment will become saturated, thereby preventing voice traffic to a given area. Such a scenario is not merely theoretical; instances of this contention have been well documented [78, 72, 50, 120, 136, 98].

In order to determine the required number of messages to induce saturation, the details of the air interface must be examined. While the following analysis of this vulnerability focuses on GSM networks, other systems (e.g. CDMA [172]) appear to be vulnerable to similar attacks.

The GSM air interface is a timesharing system. This technique is commonly employed in a variety of systems to provide an equal distribution of resources between multiple parties. Each channel is divided into eight timeslots and, when viewed as a whole, form a frame. During a given timeslot on a *Transmission Channel* (TRX), the assigned user receives full control of the channel. From the telephony perspective, a user assigned to a given TCH is able to transmit voice data once per frame. In order to provide the illusion

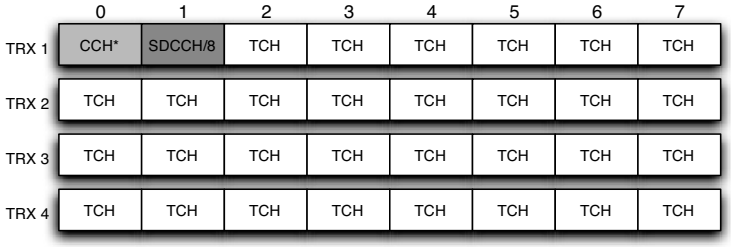

Fig. 5.4. An example air interface with four carriers (each showing a single frame). The first time slot of the first carrier is the Common CCH. The second time slot of the first channel is reserved for SDCCH connections. Over the course of a multiframe, capacity for eight users is allotted. The remaining time slots across all carriers are designated for voice data. This setup is common in many urban areas.

of continuous voice sampling, the frame length is limited to 4.615 ms. An illustration of this system is shown in Figure 5.4.

Because the bandwidth within a given frame is limited, data (especially relating to the CCH) must often span a number of frames, as depicted in Figure 5.5. This aggregation is known as a multiframe and is typically comprised of 51 frames[7]. For example, over the course of a single multiframe, the base station is able to dedicate up to 34 of the 51 Common CCH slots to paging operations.

Each channel has distinct characteristics. While the PCH is used to signal each incoming call and text message, its commitment to each session is limited to the transmission of a TMSI. TCHs, on the other hand, remain occupied for the duration of a call, which on average is a number of minutes [131]. The SDDCH, which has approximately the same bandwidth as the PCH across a multiframe, is occupied for a number of seconds per session establishment. Accordingly, in many scenarios, this channel can become a bottleneck.

In order to determine the characteristics of the wireless bottleneck, it is necessary to understand the available bandwidth. As shown in Figure 5.5, each SDCCH spans four logically consecutive timeslots in a multiframe. With 184 bits per control channel unit and a multiframe cycle time of 235.36 ms, the effective bandwidth is 782 bps [16]. Given that authentication, TMSI renewal, the enabling of encryption, and the 160 byte text message must be transferred, a single SDCCH is commonly held by an individual session for between four and five seconds [131]. The gray-box testing in Section 5.3 reinforces the plausibility of this value by observing no messages delivered in under six seconds.

This service time translates into the ability to handle up to 900 SMS sessions per hour on each SDCCH. In real systems, the total number of SDCCHs

[7] Multiframes can actually contain 26, 51 or 52 frames. A justification for each case is available in the standards [16].

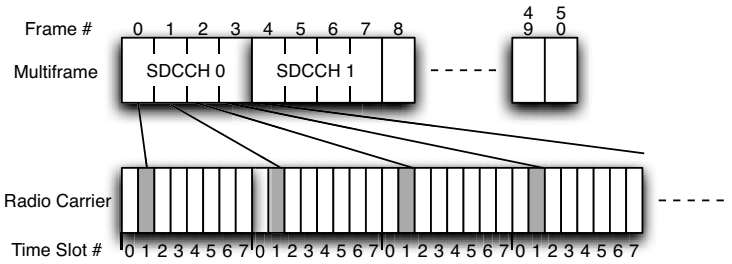

Fig. 5.5. Timeslot 1 from each frame in a multiframe creates the logical SDCCH channel. In a single multiframe, up to eight users can receive SDCCH access.

available in a sector is typically equal to twice the number of carriers[8], or one per three to four voice channels. For example, in an urban location such as the one demonstrated in Figure 5.4 where a total of four carriers are used, a total of eight SDCCHs are allocated. A less populated suburban or rural sector may only have two carriers per area and therefore have four allocated SDCCHs. Densely populated metropolitan sectors may have as many as six carriers and therefore support up to 12 SDCCHs per area.

We now calculate the maximum capacity of the system for an area. As indicated in a study conducted by the *National Communications System* (NCS) [131], the city of Washington D.C. has 40 cellular towers and a total of 120 sectors. This number reflects sectors of approximately 0.5 to 0.75 mi^2 through the 68.2 mi^2 city. Assuming that each of the sectors has eight SDCCHs, the total number of messages per second needed to saturate the SDCCH capacity C is:

$$\begin{aligned} C &\simeq (120 \text{ sectors}) \left(\frac{8 \text{ SDCCH}}{1 \text{ sector}}\right) \left(\frac{900 \text{ msgs/hr}}{1 \text{ SDCCH}}\right) \\ &\simeq 864,000 \text{ msgs/hr} \\ &\simeq 240 \text{ msgs/sec} \end{aligned}$$

Manhattan is smaller in area at 31.1 mi^2. Assuming the same sector distribution as Washington D.C., there are 55 sectors. Due to the greater population density, we assume 12 SDCCHs are used per sector.

[8] Actual allocation of SDCCH channels may vary across implementations; however, these are the generally accepted values throughout the community.

Area	# Sectors	# SDCCHs per sector	SMS Capacity	Upload Bandwidth*	Multi-Recipient Bandwidth*
Washington D.C. ($68.2\ mi^2$)	120	8	240 msgs/sec	2812.5 kbps	281.25 kbps
		12	360 msgs/sec	4218.8 kbps	421.88 kbps
		24	720 msgs/sec	8437.5 kbps	843.75 kbps
Manhattan ($31.1\ mi^2$)	55	8	110 msgs/sec	1289.1 kbps	128.91 kbps
		12	165 msgs/sec	1933.6 kbps	193.66 kbps
		24	330 msgs/sec	3867.2 kbps	386.72 kbps

* assuming 1500 bytes per message

Table 5.2. Required upload bandwidth to saturate an empty network

$$
\begin{aligned}
C &\simeq (55 \text{ sectors}) \left(\frac{12 \text{ SDCCH}}{1 \text{ sector}}\right) \left(\frac{900 \text{ msg/hr}}{1 \text{ SDCCH}}\right) \\
&\simeq 594,000 \text{ msg/hr} \\
&\simeq 165 \text{ msg/sec}
\end{aligned}
$$

Given that SMSCs in use by service providers in 2002 were capable of processing 20,000 messages per second [29], such volumes are achievable even in the *hypothetical* case of a sector having twice this number of SDCCHs.

Using a source transmission size of 1500 bytes to submit an SMS from the Internet, Table 5.2 shows the bandwidth required at the source to saturate the control channels, thereby incapacitating legitimate voice and text messaging services for Washington D.C. and Manhattan. The adversary's bandwidth requirements can be reduced by an order of magnitude when attacking providers including Verizon and Cingular Wireless due to the ability to have a single message repeated to up to ten recipients.

As mentioned in Section 5.3.1, sending this magnitude of messages to a small number of recipients would degrade the effectiveness of such an attack. Targeted phones would quickly see their buffers reach capacity. Undeliverable messages would then be buffered in the network until the space allotted per user was also exhausted. These accounts would likely be flagged and potentially temporarily shut down for receiving a high number of messages in a short period of time, thereby fully extinguishing the attack. Clever usage of well constructed hit-lists keeps the number of messages seen by individual phones far below realistic thresholds for rate limitation on individual targets.

Using the conservative population and demographic numbers cited from the NCS technical bulletin [131][9] and assuming 50% of the wireless subscribers in Washington are serviced by the same network, an even distribution of messages would require the delivery of approximately 5.04 messages to each phone per hour (1 message every 11.92 minutes) to saturate Washington D.C. If the

[9] 572,059 people with 60% wireless penetration and 8 SDCCHs (and that devices are powered on).

percentage of subscribers receiving service from a provider is closer to 25%, the number is only 10.07 messages per hour (1 message every 5.96 minutes). In a more densely populated city such as Manhattan, with a population estimated at 1,318,000 with 60% wireless penetration and 12 SDCCHs, only 1.502 messages would have to be received per user per hour if half of the wireless clientele use the same provider. That number increases slightly to 3.01 if the number is closer to 25%.

Depending on the intended duration of an attack, the creation of very large hit-lists may not be necessary. An adversary may only require a five minute service outage to accomplish their mission. Assuming that the attacker created a hit-list with only 2500 phone numbers, with each target having a buffer of 50 messages and launched their attack in a city with 8 SDCCHs (e.g., Washington D.C.), uniform random use of the hit-list would deliver a single message to each phone every 10.4 seconds, allowing the attack to last 8.68 minutes before buffer exhaustion. Similar to the most dangerous worms in the Internet, this attack could be completed before anyone capable of thwarting it could respond.

When compared to the requisite bandwidth to launch these attacks listed in Table 5.2, many of these scenarios can be executed from a single high-end cable modem. A more distributed, less bandwidth intense attack might instead be launched from a *small* zombie network or from a number of compromised mobile phone or from a number of compromised mobile phones.

While the disruption of voice and SMS service is achievable through frequency jamming attacks, physical proximity to a target is required. The danger of the attacks discussed in this chapter is their ability to be launched from any point on the globe without the perpetrator ever having been present in the targeted area.

5.5.3 Regional Service

Both popularity and the potential for high revenue have forced service providers to investigate methods of increasing SMS capacity in their networks. Already, a number of major industrial players [51, 87] offer solutions designed to offload SMS traffic from the traditional SS7 phone system onto less expensive, higher bandwidth IP-based networks. New SMSCs, each capable of processing some 20,000 SMS messages per second, would help to quickly disseminate the constantly increasing demand.

Advanced services including *General Packet Radio Service* (GPRS) and *Enhanced Data rates for GSM Evolution* (EDGE) promise high speed data connections to the Internet for mobile devices. While offering to alleviate multimedia traffic at the SMSC and potentially send some SMS messages, these data services are widely viewed as complimentary to SMS and will thus not replace SMS's functionality in the foreseeable future [45][10]. In terms of

[10] SMS over GPRS is already in service; however, it is not the default method of SMS delivery on GPRS-capable phones and must be activated by the user. Fur-

SMS delivery, all aspects of the network are increasing available bandwidth except the SDCCH bottleneck. A discussion of vulnerabilities in these higher bandwidth services is given in Chapter 6.

We examine a conservative attack on the cellular infrastructure in the United States. From the United States Census in 2000, approximately 92,505 mi^2[188] are considered urban. This 2.62% of the land is home to approximately 80% of the nation's population. We first model the attack by assuming that all urban areas in the country have high-capacity sectors (8 SDCCHs per sector). This assumption leads to the results shown below:

$$\begin{aligned} C &\simeq \left(\frac{8 \text{ SDCCH}}{1 \text{ sector}}\right)\left(\frac{900 \text{ msg/hr}}{1 \text{ SDCCH}}\right)\left(\frac{1.7595 \text{ sectors}}{1 \text{ mi}^2}\right) \\ &\quad (92,505 \text{ mi}^2) \\ &\simeq 1,171,890,342 \text{ msg/hr} \\ &\simeq 325,525 \text{ msg/sec} \end{aligned}$$

This attack would require approximately 3.8 Gbps and a nation-wide hitlist to be successful. If the adversary is able to submit a single message to up to ten different recipients, the requisite bandwidth for the attacker drops to approximately 370 Mbps. Considering that previous *Distributed Denial of Service* (DDoS) attacks have crippled websites such as Yahoo! with gigabits per second bandwidth, an attack on the entire cellular infrastructure is within the capability of sophisticated adversaries.

5.6 Network Characterization

The calculations in the previous section offer only coarse-grained approximations of targeted text messaging attacks. For instance, the subtle interplay of realistic traffic patterns with a host of network components are not captured. Without unrestricted access to an operational cellular network, characterizing the behavior of specific components is difficult. In order to accurately characterize the impact of such an attack on the air interface, we have developed a detailed GSM simulator. We base our system behavior and parameter settings on publicly available standards documents [17, 16].

A cellular deployment similar to that found in Manhattan [131] is used as our baseline scenario. Each of the 55 sectors in the city has 12 SDCCHs[11]. We assume both call requests and text messages arrive with a Poisson distribution

thermore, SMS over GPRS still defaults to the standard SMS delivery mechanism when GPRS is unavailable

[11] In reality, only the highest capacity sectors would be so over-provisioned [131], making this a conservative estimate for every sector in a city.

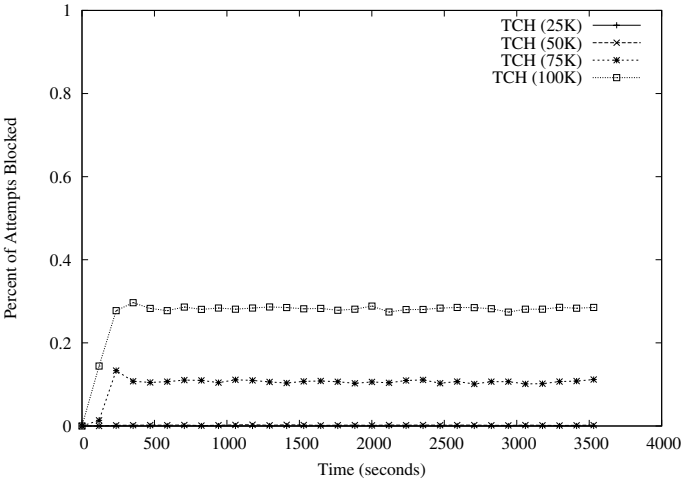

Fig. 5.6. Blocking characteristics of a network under a variety of traffic intensities.

and that TCH and SDCCH holding times are exponentially distributed around the appropriate means (see Table 5.3) unless explicitly stated otherwise. Such values are well within standard operating conditions [118, 125, 119].

Figure 5.6 illustrates the blocking rates for traffic channels under four different voice traffic loads. Most relevant to the current discussion is the nonexistence of call blocking. The absence of such blocking reinforces the robustness of the design of GSM as a voice communication system. Specifically, the only points of congestion in the system are the traffic channels themselves. Figure 5.7 further supports the blocking data by illustrating very low SDCCH utilization rates for offered loads of both 50 and 100K $calls/hour$.

Elevated loads may represent significant public gatherings (e.g., concerts, celebrations), holiday spikes or large-scale emergencies. Blocking on other channels begins to become observable only under such extreme circumstances. Figure 5.8 highlights an emergency situation in which the call and SMS rate

Table 5.3. System and Attack Parameters

μ_{TCH}^{-1}	120 sec [135]
$\mu_{SDCCH,call}^{-1}$	1.5 sec [135]
$\mu_{SDCCH,SMS}^{-1}$	4 sec [131]
λ_{call}	50,000 calls/city/hr .2525 calls/sector/sec
$\lambda_{SMS,attack}$	495 msgs/city/sec 9 msgs/sector/sec
$\lambda_{SMS,regular}$	138.6K/city/hr 0.7 msgs/sector/sec

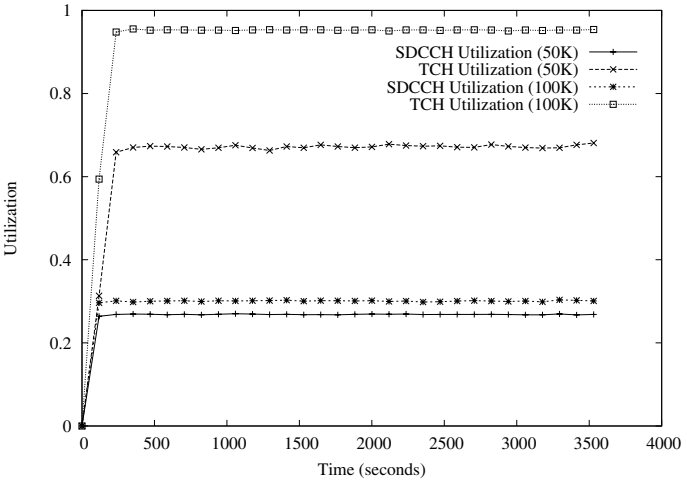

Fig. 5.7. Channel utilization characteristics of a network under a variety of traffic intensities.

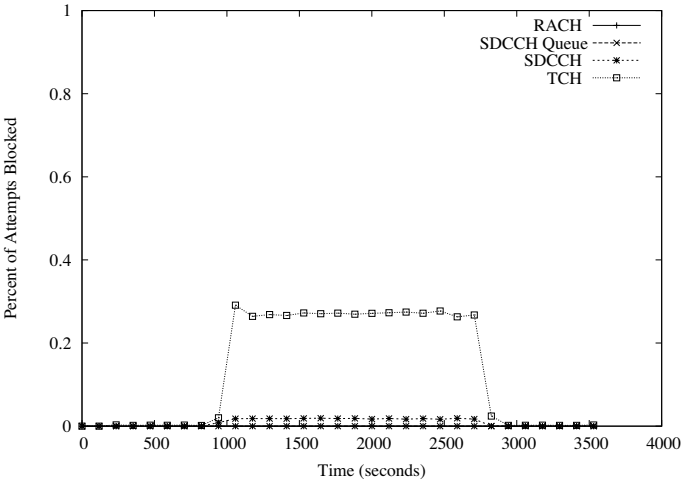

Fig. 5.8. Blocking characteristics of a network under emergency conditions (100K calls/hr, 276K msgs/hour).

spikes from 50K *calls/hour* to 100K *calls/hour* and 138K *msgs/hour* to 276K *msgs/hour*, respectively. Figure 5.9, which shows the channel utilization data for the "emergency" scenario, reinforces that it is only under extreme duress that other channels in the system begin to saturate.

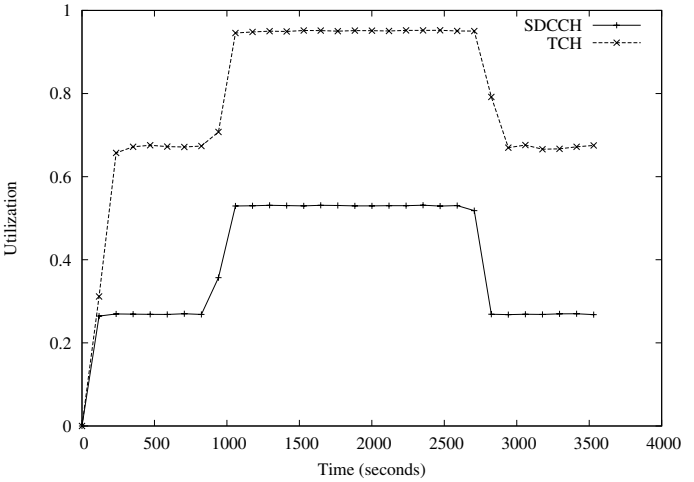

Fig. 5.9. Channel utilization characteristics of a network under emergency conditions (100K calls/hr, 276K msgs/hour).

5.7 Attack Characterization

In order to judge the efficacy of any countermeasure against targeted SMS attacks, it is necessary to fully characterize such an event. We seek to understand the observed conditions and the subtle interplay of network components given a wide range of inputs. We use the simulator described in the previous subsection and the parameters in Table 5.3 to understand such attacks.

To isolate the impact of blocking caused by SDCCH congestion, we do not include SDCCH queues; we examine the impact of such queues in Section 5.9. If a call request or text message arrives when all SDCCHs are occupied, the request is blocked.

A sector is observed for a total of 60 minutes, in which the middle 30 minutes are exposed to a targeted SMS attack. The SMS attack intensity is varied between 4 and 13 times the normal SMS load, i.e., $\lambda_{SMS} = 165\ msgs/sec$ (3 messages/second/sector) to $\lambda_{SMS} = 495\ msgs/sec$ (9 messages/second/sector)[12]. All results are the average of 1000 runs, each using randomly generated traffic patterns consistent with the above parameters.

Because delay variability is likely throughout the network, and because SDCCH holding times will not be deterministic due to varying processing times and errors on the wireless links, the perfect attack presented in our previous work would be difficult to achieve in real networks. Accordingly, we investigate a number of flow arrival characteristics while considering exponentially

[12] Because DoS attacks on the Internet frequently exhibit more than an entire year's volume of traffic [150], such an increase is relatively insignificant.

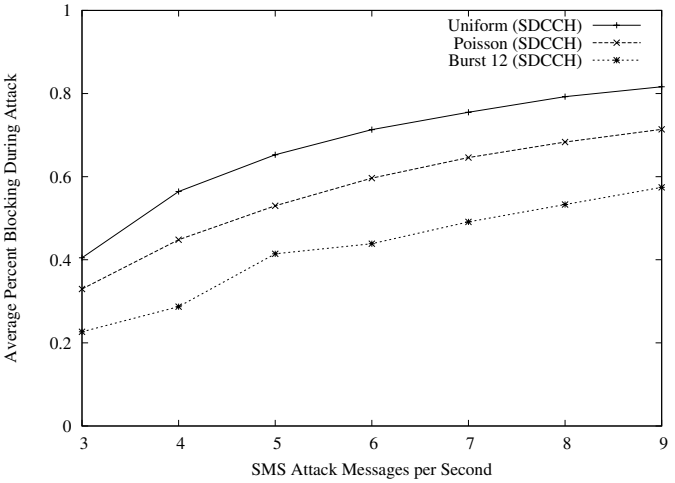

Fig. 5.10. The blocking probability for traffic exhibiting uniform, Poisson and bursty arrival patterns over varying attack strengths.

distributed SDCCH holding times. Figure 5.10 illustrates the effectiveness of attacks when messages arrive with a Poisson, bursty (12 messages delivered back-to-back), or uniform distribution. Notice that, due to the addition of variability, bursty attacks are the least successful of the three. This is because it is unlikely that 12 text messages arriving back-to-back will all find unoccupied SDCCHs. Thus, blocking occurs on the attack messages, and legitimate traffic that arrives between bursts has a higher probability of finding an available SDCCH. The most effective attack is when messages arrive uniformly spaced; however, due to variable network delay, such an attack would also be difficult to realize.

Our remaining experiments therefore assume a Poisson distribution for the arrival of text messages. We use an attack intensity of 495 $msgs/sec$, which is equal to 9 messages/second/sector and yields a blocking probability of 71%. For this case we show the SDCCH and TCH utilization in Figure 5.11. This figure shows the effectiveness of the attack: during the attack, the SDCCH utilization is near 1.0, and the TCH utilization drops from close to 70% down to approximately 20%. This shows that although TCHs are available for voice calls, they cannot be allocated due to SDCCH congestion. Our experiments suggest that, at this rate, no other bottlenecks in the system exist, including other control channels or the SS7 signaling links.

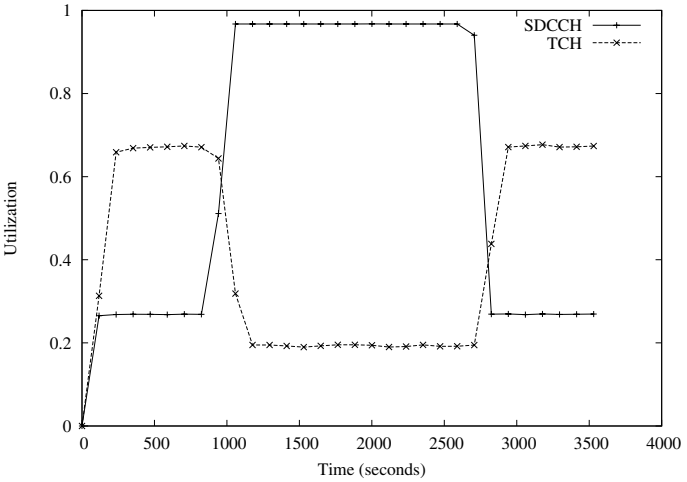

Fig. 5.11. The utilization of SDCCHs and TCHs for an attack exhibiting a Poisson interarrival at a rate of 495 messages/second.

5.8 Current Solutions

Voice communications have traditionally received priority in telecommunications systems. Because voice has been the dominant means by which people interact via these networks, providers allow for the degradation of other services in order to achieve high availability for the voice services. There are, however, an increasing set of scenarios in which the priority of services begins to change.

On September 11th, 2001, service providers experienced significant surges in usage. Verizon Wireless reported the number of calls made increased by more than 100% above average levels. Cingular Wireless experienced an increase of over 1000% for calls bound for the greater Washington D.C area [131]. In spite of the increased call volume, SMS messages were still received in even the most inundated areas because the control channels used for their delivery remained uncongested. In both emergency and day-to-day situations, the utility of text messaging has increased to the same level as voice communications for significant portions of the population [4].

A simple analysis shows that repurposing all resources used in voice telephony for SMS delivery would greatly increase overall network throughput. Specifically, more users would be able to communicate concurrently under such a scheme. However, regulations constraining the availability of voice services during times of emergency prevent such an approach from being implemented. Given requirements for reliable voice and an ever-increasing demand for and reliance upon SMS, mitigation strategies must not only maintain the avail-

ability of voice services, but also maximize the throughput of legitimate text messages.

Cellular providers have introduced a number of mitigation solutions into phone networks to combat the SMS-based DoS attacks. These solutions focus on *rate limiting* the source of the messages and are ineffective against all but the least sophisticated adversary. To illustrate, the primary countermeasure discovered during our gray-box testing in Section 5.3 was a per-source volume restriction at the SMS gateway. Such restrictions would, for example, allow only 50 messages from a single IP address. The ability to spoof IP addresses and the existence of bot networks render this solution impotent. Another popular deployed solution filters SMS traffic based on the textual content. Similar to SPAM filtering, this approach is effective in eliminating undesirable traffic only if the content is predictable. However, an adversary can bypass this countermeasure by generating legitimate looking SMS traffic from randomly generated simple texts, e.g., "*Remember to pick up your shirts from the dry cleaner on your way home. -Alice*". As a proof of concept, we used text samples from provider "Terms of Service" documents during our gray-box testing. None of these randomly chosen strings caused any message to be filtered.

Note that these and the overwhelming majority of other solutions deployed in response to the SMS vulnerability can be classified as *edge solutions*[13]. Ineffective by construction because of their lack of context, such solutions try to regulate the traffic flowing from the Internet into the provider network at its edge. Limiting the total volume of traffic coming across all interfaces results simply in reduced income under normal operating conditions. For example, a total of 1,000 email-generated text messages per second distributed across a nation cause no ill effects to the network and generates significant revenue. As is shown in Section 5.7, such a volume of traffic is more than sufficient to deny service to Manhattan. If a provider were to limit the number of messages to a much smaller number (e.g., 250 messages per second from the Internet), the adversary could simply tighten their targeted attack area so that this volume of messages continued to elicit the same effect. Moreover, an adversary compromising a large number of mobile phones could bypass such edge solutions completely as the source of the attack would be located within the network.

Rate limitation is largely unattractive even within the core network. The distributed nature of SMSCs, through which all text messages flow, makes it difficult to coordinate real-time filtering in response to targeted attacks. In addition, because provider networks cover huge geographic areas and consist of hundreds of thousands of network elements, any compromised element can be a conduit for malicious traffic. If left unregulated, the connections between provider networks can also be exploited to inject SMS traffic.

It is therefore prudent to assume that an adversary is able to successfully submit a large number of text messages into a cellular network. The

[13] This class of solutions is designed to protect the "edge" of the network

Table 5.4. Commonly Used Variables

λ_{call}	Arrival rate of voice calls
λ_{SMS}	Arrival rate of text messages
$\mu_{SDCCH,call}$	Service rate of voice calls at SDCCH
$\mu_{TCH,call}$	Service rate of voice calls at TCH
$\mu_{SDCCH,SMS}$	Service rate of text messages at SDCCH
ρ_{call}	Call traffic intensity
ρ_{SMS}	SMS traffic intensity

defenses discussed in the following sections are dedicated to protecting the resource that is being exploited in the SMS attack – the bandwidth constrained SDCCHs. Note that the Internet faces a similar conundrum: once dominant perimeter defenses are failing in the face of dissolving network borders, e.g., as caused by wireless connectivity and larger and more geographically distributed networks [114]. While we certainly recognize that edge solutions provide some barrier to attack by filtering out obvious SPAM, the approaches below provide a "defense in depth" approach to protect telecommunications networks.

5.9 Queue Management

While the solutions currently in place are not sufficient to prevent targeted text messaging attacks, the application of other well-known techniques has great potential. A variety of queue management mechanisms, for example, have been extensively investigated and tested in IP networks. As a first step towards mitigating targeted text messaging attacks, we apply variants of two of the more prominent techniques from this area – Weighted Fair Queuing and Weighted Random Early Discard. Because the environment in which these solutions are applied is significantly different from an IP network, the behavior of these solutions is not entirely as expected.

To assist those readers not familiar with queuing theory, Table 5.4 provides definitions for the variables used throughout the remainder of this chapter.

5.9.1 Weighted Fair Queuing

Analysis

Because we cannot rely on rate limitation at the source of messages, we now explore network-based solutions. Fair Queuing [129] is a scheduling algorithm that separates flows into individual queues and then apportions bandwidth equally between them. Designed to emulate bit-wise interleaving, Fair Queuing services queues in a round-robin fashion. Packets are transmitted when their calculated interleaved finishing time is the shortest. Building priority into such a system is a simple task of assigning weights to flows. Known as *Weighted*

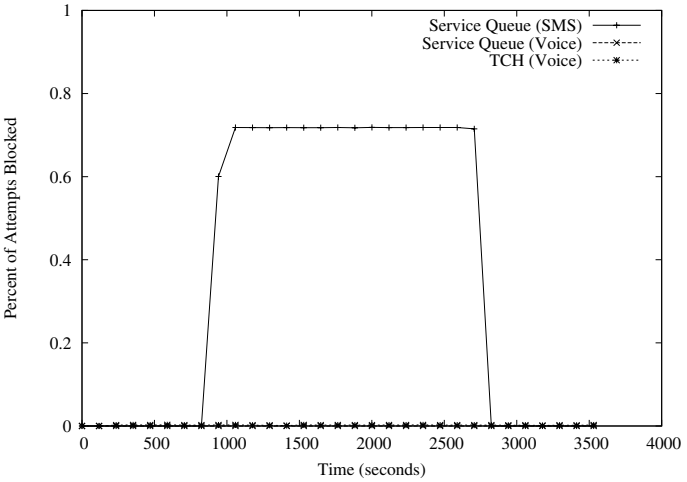

Fig. 5.12. The simulated blocking probability for a sector implementing WFQ. Notice that voice calls are unaffected by the attack, whereas the majority of text messages are dropped.

Fair Queuing (WFQ) [57], this technique can be used to give incoming voice calls priority over SMS.

We provide a simplified analysis to characterize the performance of WFQ in this scenario. We apply WFQ to the service queues of the SDDCH. We create two waiting queues, one for voice requests and one for SMS requests, respectively. The size of the call queue is 6 and the size of the SMS queue is 12 in order to buffer small bursts but to limit call processing latency. To determine the relative blocking probability and utilization of the voice and SMS flows, we begin by assuming the conditions set forth in Tables 5.4 and 5.3.

WFQ can be approximated as a *General Processor Sharing* system (GPS) [162]. The average service rate of such systems is the weighted average of the service rates of all classes of service requests. In our case we have two types of requests: voice calls with $\lambda_{call} = 0.2525$ $calls/sector/sec$ and an average service time on the SDCCH of $\mu_{call}^{-1} = 1.5$ seconds, and SMS requests with $\lambda_{SMS} = 9.7$ $msgs/sector/sec$ (attack traffic + regular traffic) and $\mu_{SMS}^{-1} = 4$ seconds. Therefore, for our system, $\mu^{-1} = 3.94$ seconds.

Although our system has multiple servers (SDCCHs), and is thus an M/M/n system, because it is operating at high loads during an attack, it may be approximated by an M/M/1 system with its $\mu = n\mu^{'}$, where $\mu^{'}$ is the service rate calculated above. Using these values, and accounting for the weighting of 2:1 for servicing call requests, the call traffic intensity $\lambda_{call}/\mu_{call} = \rho_{call} = 0.04$, and the expected call queue occupancy is about 1%. Because the ρ_{SMS} is much greater than 1, its SMS queue occupancy is

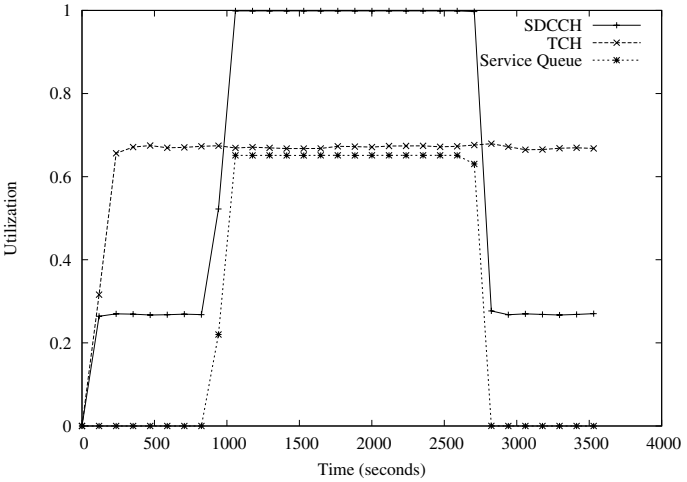

Fig. 5.13. Utilization for a sector implementing WFQ. Note that TCH utilization is constant throughout the attack.

approximately 100%. When combined, the total queue occupancy is approximately 67%.

Simulation

Buffering alone is not sufficient to protect against congestion [129, 99]; rather, mechanisms designed to mediate tail drop blocking are necessary. We apply WFQ with a weight of 2 for calls and 1 for SMS messages to ensure voice calls receive a suitable amount of SDCCH bandwidth.

Figure 5.12 illustrates the resulting blocking for a sector implementing WFQ. The preferential treatment of voice traffic eliminates the blocking previously seen in an unprotected system. Incoming text messages, however, continue to experience roughly the same blocking (72%) observed by all traffic in the base attack scenario. As is shown in Figure 5.13, the queue itself does nothing to prevent congestion. Total queue utilization is 65%. As two-thirds of the queue space is available to text messaging, this represents a near total average occupancy of the SMS queue and a virtually unused voice traffic queue. Such an observation conforms to our analytical results. This figure also demonstrates the ability to protect voice services, as TCH utilization is not lowered during the attack.

The advantage to implementing the WFQ mechanism is not only its relative simplicity, but also its effectiveness in preventing degradation of voice services during targeted SMS attacks. Unfortunately, the granularity for prioritizing text messages is insufficient to provide adequate service to those users

relying upon text messaging as their dominant means of communication. Accordingly, if users believe that their traffic is unlikely to be delivered, their faith in text messaging as a reliable service will decrease. While finer granularity can be provided by adding one queue per SMS class, this solution will result in inefficient memory use and complexity. We discuss means of adding such granularity through the use of WRED.

5.9.2 Weighted Random Early Detection

Analysis

Active queue management has received a great deal of attention as a congestion avoidance mechanism in the Internet. *Random Early Detection* (RED) [71, 44], one of the better known techniques from this field, is a particularly effective means of coping with potentially damaging quantities of text messages. While traditionally used to address TCP congestion, RED helps to prevent queue lockout and RED drops packets arriving to a queue with a probability that is a function of the weighted queue occupancy average, Q_{avg}. Packets arriving to a queue capacity below a threshold, t_{min}, are never dropped. Packets arriving to a queue capacity above some value t_{max} are always dropped. Between t_{min} and t_{max}, packets are dropped with a linearly increasing probability up to $P_{drop,max}$. This probability, P_{drop}, is calculated as follows[14]:

$$P_{drop} = P_{drop,max} \cdot (Q_{avg} - t_{min})/(t_{max} - t_{min}) \tag{5.1}$$

The advantages to this approach are twofold: first, lockout becomes more difficult as packets are purposefully dropped with greater frequency; secondly, because the capacity of busy queues stays closer to a moving average and not capacity, space typically exists to accommodate sudden bursts of traffic. However, one of the chief difficulties with traditional RED is that it eliminates the ability of a provider to offer QoS guarantees because all traffic entering a queue is dropped with equal probability. *Weighted Random Early Detection* (WRED) solves this problem by basing the probability a given incoming message is dropped on an attribute such as its contents, source or destination. Arriving messages not meeting some priority are therefore subject to increased probability of drop. The dropping probability for each class of message is tuned by setting $t_{priority,min}$ and $t_{priority,max}$ for each class.

We consider the use of authentication as a means of creating messaging priority classes. For example, during a crisis, messages injected to a network from the Internet by an authenticated municipality or from emergency personnel could receive priority over all other text messages. A number of municipalities already use such systems for emergency [155] and traffic updates [170]. Messages from authenticated users within the network itself receive secondary priority. Unauthenticated messages originating from the Internet are delivered

[14] Some variants of RED additionally incorporate a *count* variable. Equation 5.1 is the simplest version of RED defined by RFC 2309 [44].

with the lowest priority. Such a system would allow the informative messages (i.e., evacuation plans, additional warnings) to be quickly distributed amongst the population. The remaining messages would then be delivered at ratios corresponding to their priority level. We assume that packet priority marking occurs at the SMSCs such that additional computational burden is not placed on base stations.

Here, we illustrate how WRED can provide differentiated service to different classes of SMS traffic using the attack scenario described in Tables 5.3 and 5.4. We maintain separate queues, which are served in a round robin fashion, for voice requests and SMS requests. We apply WRED to the SMS queue. We assume legitimate text messages arrive at a sector with an average rate of 0.7 $msgs/sec$ with the following distribution: 10% high priority, 80% medium priority, and 10% low priority. The attack generates an additional 9 $msgs/sec$.

To accommodate sudden bursts of high priority SMS traffic, we choose an SMS queue size of 12. Because we desire low latency delivery of high priority messages, we target an average queue occupancy $Q_{avg} = 3$.

To meet this objective, we must set $t_{low,min}$ and $t_{low,max}$. For M/M/n systems with a finite queue of size m, the number of messages in the queue, N_Q, is:

$$N_Q = P_Q \frac{\rho}{1-\rho} \tag{5.2}$$

where:

$$P_Q = \frac{p_0(m\rho)^m}{m!(1-\rho)} \tag{5.3}$$

where:

$$p_0 = \left[\sum_{n=0}^{m-1} \frac{(m\rho)^n}{n!} + \frac{(m\rho)^m}{m!(1-\rho)} \right]^{-1} \tag{5.4}$$

Setting $N_Q = 3$, we derive a target load $\rho_{target} = 0.855$. ρ_{target} is the utilization desired at the SDCCHs. Thus, the packet dropping caused by WRED must reduce the actual utilization, ρ_{actual} or $\lambda_{SMS}/(\mu_{SMS} \cdot n)$, caused by the heavy offered load during an attack, to ρ_{target}. Therefore:

$$\rho_{target} = \rho_{actual}(1 - P_{drop}) \tag{5.5}$$

where P_{drop} is the overall dropping probability of WRED. For traffic with average arrival rate of $\lambda_{SMS} = 9.7$ $msgs/sec$, $\rho_{actual} = 3.23$. Solving for P_{drop},

$$P_{drop} = 1 - \frac{\rho_{target}}{\rho_{actual}} = 0.736 \tag{5.6}$$

P_{drop} can be calculated from the dropping probabilities of the individual classes of messages by ($\lambda_{low} = 9.07$):

$$P_{drop} = \frac{P_{drop,high} \cdot \lambda_{high} + P_{drop,med} \cdot \lambda_{med} + P_{drop,low} \cdot \lambda_{low}}{\lambda_{SMS}} \tag{5.7}$$

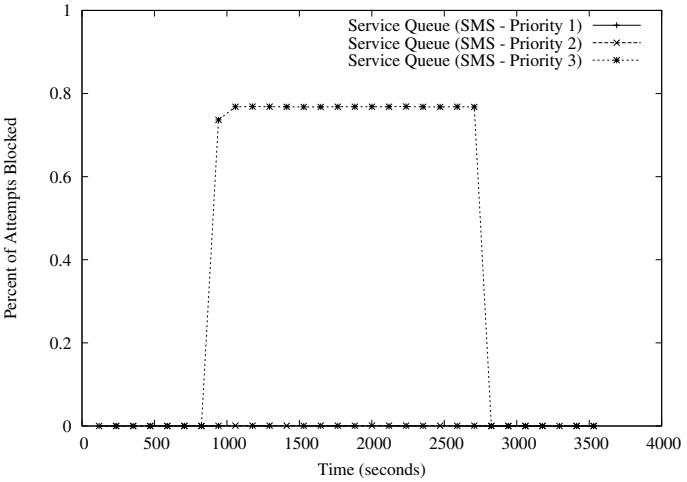

Fig. 5.14. The simulated blocking probability for a sector implementing WRED. Unlike WFQ, only Internet-originated text messages are dropped at an elevated frequency.

Because we desire to deliver all messages of high and medium priority, we set $P_{drop,high} = P_{drop,med} = 0$. Using Equation 5.7, we find $P_{drop,low} = 0.787$. This value is then used in conjunction with Equation 5.1 to determine $t_{low,min}$ and $t_{low,max}$.

The desired average queue occupancy, Q_{avg}, is 3. From equation 5.1, $t_{low,min}$ must be an integer less than the average queue occupancy. This leaves three possible values for $t_{low,min}$: 0, 1, and 2. The best fit is found when $t_{low,min} = 0$ and $t_{low,max} = 4$, resulting in 75% dropping of low priority traffic.

Using this method it is possible to set thresholds to meet delivery targets. Of course, depending on the intensity of an attack, it may not be possible to meet desired targets according to Equation 5.7, i.e., it may not be possible to limit blocking to only low priority traffic. While the method outlined here provides just an approximate solution, given the quantization error in setting $t_{low,min}$ and $t_{low,max}$ (they must be integers), we believe the method is sufficient.

Simulation

The use of a prioritized dropping policy allows a system to offer similar prioritization to WFQ while maintaining only a single queue. In our implementation of WRED, we maintain one queue for voice requests (size of 6) and one queue for SMS messages (size 12) and apply WRED to the SMS queue. We differentiate the SMS traffic by setting different thresholds for each class. We

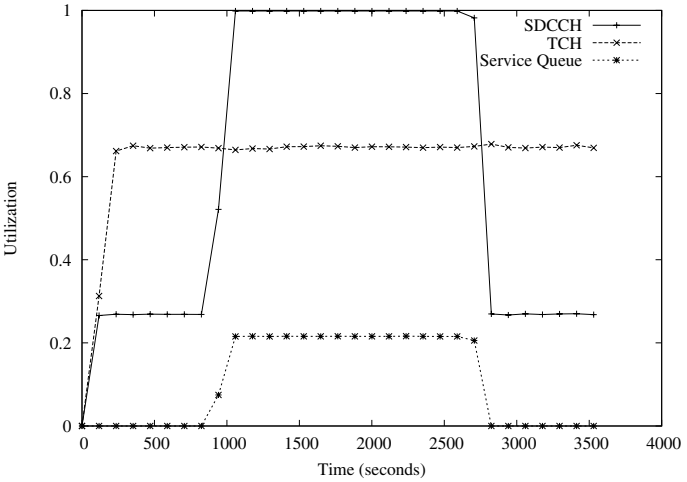

Fig. 5.15. Utilization for a sector implementing WRED. Note that the queue occupancy is low due to the decreased priority of Internet-originated messages.

assume that SMS traffic is marked upstream as having high, medium, or low priority. We assign the thresholds as ($t_{high,max} = t_{high,min} = 12$), medium ($t_{med,max} = 10, t_{med,min} = 6$) and low ($t_{low,max} = 4, t_{low,min} = 0$) priority. These priorities correspond directly to emergency priority users, network customers and Internet-originated messages, respectively. Q_{avg} is maintained as a simple weighted average with a weight of 0.8 on the most recent sample.

Figure 5.14 gives the blocking for each of the three priorities of text messages. Because voice calls never block in these simulations, we omit them from this graph. Both high and medium priority flows also do not experience blocking throughout the simulations. The blocking of Internet-originated messages averages 77%, approximately the same blocking probability experienced by all incoming messages in the base attack scenarios. Service queue utilization, shown in Figure 5.15, is 20%. With a total queue capacity of 18, this corresponds to an average occupancy of 3.88 messages. Also notice that the TCH occupancy is maintained throughout the attack.

The parameters used in this simulation are the same as those in Table 5.3. We set the medium priority thresholds to allow some loss at very high loads to protect the high priority traffic under extreme circumstances, but because our average queue occupancy is about 3.9, no dropping of medium priority messages occurs. This matches well with our analytical results.

Systems implementing WRED not only match the elimination of voice call blocking seen through the use of WFQ, but also offer significantly improved performance in terms of message delivery. Implementing this solution, however, faces its own challenges. The authentication of high priority messages,

for example, would require the use of additional infrastructure. High priority messages originating outside the network, such as emergency messages distributed by a city, may require the use of a dedicated line and/or the use of a public key infrastructure (PKI) for authentication. Because of historical difficulties effectively achieving the latter [59], implementing such a system may prove difficult. Even with such protections, this mechanism fails to protect the system against insider attacks. If the machine responsible for sending high priority messages into the network or user phones are compromised by malware, systems implementing WRED lose their messaging performance improvements over the WFQ solution. Note that networks not bounding priority to specific geographic regions can potentially be attacked through any compromised high priority device.

5.9.3 Summary

As demonstrated in this section, queue management techniques are a valuable tool for protecting voice telephony from targeted SMS attacks. Whereas both approaches protect voice from targeted SMS attacks, WRED allows for greater classification of traffic without the significant added overhead WFQ would require to implement multiple queues. The benefits of these schemes, however, are more limited than in traditional data networks. For instance, the shedding of TCP packets in an IP network should cause flow control mechanisms to reduce transmission rates. Because the same does not happen here, queue management techniques essentially "bail water" until an attack subsides. Accordingly, such mechanisms should be relied upon as a last line of defense. We therefore look to techniques that reapportion the resources on the air interface for a more flexible response.

5.10 Resource Provisioning

None of the above methods deal with the system bottleneck directly; rather, they strive to affect traffic before it reaches the air interface. An alternative strategy of addressing targeted SMS attacks instead focuses on the reallocation of the available messaging bandwidth. We therefore investigate a variety of techniques that modify the way in which the air interface is used.

To analyze these techniques we resort to simple Erlang-B queuing analysis. We present a brief background here. For more details see Schwartz [162]. In a system with n servers, and an offered load in Erlangs of A, the probability that an arriving request is blocked because all servers are occupied is given by:

$$P_B = \frac{\frac{A^n}{n!}}{\sum_{l=0}^{l=n-1} \frac{Al}{l!}} \tag{5.8}$$

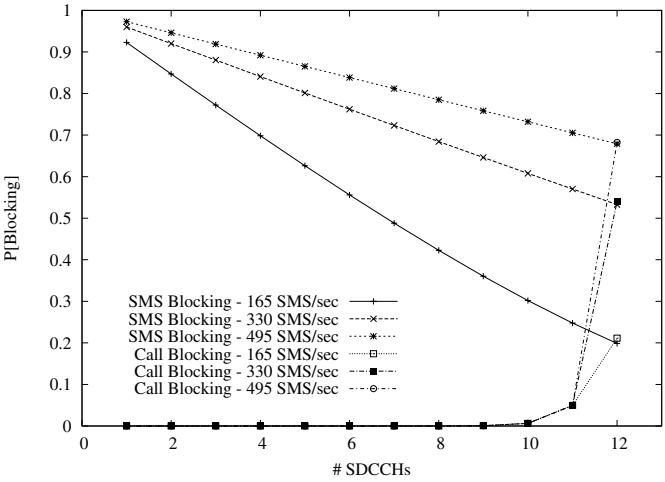

Fig. 5.16. The probability that incoming calls and SMS messages are blocked in a system implementing SRP. We vary the number of SDCCHs that will accept SMS requests from 1 to 12(all).

The load in Erlangs is the same as the utilization, ρ, in a queuing system; it is simply the offered load multiplied by the service time of the resource. The expected occupancy of the servers is given by:

$$E(n) = \rho(1 - P_B) \tag{5.9}$$

5.10.1 Strict Resource Provisioning

Analysis

Under normal conditions, the resources for service setup and delivery are over-provisioned. At a rate of $50,000$ *calls/city/hour* in our baseline scenario, for example, the calculated average utilization of SDCCHs per sector is approximately 2%. Given this observation, if a subset of the total SDCCHs can be used only by voice calls, blocking due to targeted SMS attacks can be significantly mitigated. Our first air interface provisioning technique, *Strict Resource Provisioning* (SRP), attempts to address this contention by allowing text messages to occupy only a subset of the total number of SDCCHs in a sector. Requests for incoming voice calls can compete for the entire set of SDCCHs, including the subset used for SMS. In order to determine appropriate parameters for systems using SRP, we apply Equations 5.8 and 5.9.

To isolate the effectiveness of SRP, we consider a system with no queue. Figure 5.16 shows the blocking probabilities for a system using SRP when we

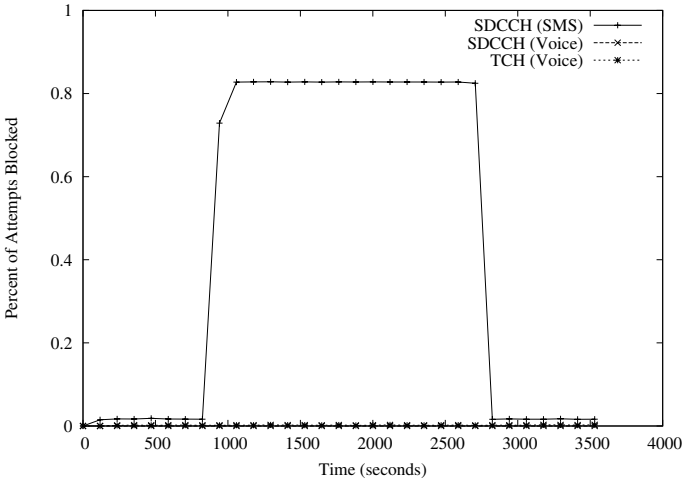

Fig. 5.17. Blocking for a sector implementing SRP

vary the number of SDCCHs that will accept SMS requests from 1 to 12 (all). Because incoming text messages only compete with voice calls for a subset of the resources, any resulting call blocking is strictly a function of the size of the subset of voice-only SDCCHs. The attacks of intensity 165, 330, and 495 $msgs/city/sec$ (3, 6, and 9 messages/second/sector) have virtually no impact on voice calls until the full complement of SDCCHs are made available to all traffic. In fact, it is not until 10 SDCCHs are made available to SMS traffic that the blocking probability for incoming voice calls reaches 1%.

By limiting the number of SDCCHs that will serve SMS requests, the blocking for SMS is increased. When only six SDCCHs are available to text messages, blocking probabilities for SMS are as high as 84%. Because significant numbers of people rely upon text messaging as their primary means of communication, such parameters should be carefully tuned.

Simulation

Before characterizing the SRP technique, careful consideration was given to the selection of operating parameters. Because many MSCs are capable of processing up to $500K$ $calls/hour$, we engineer our solution to be robust to large spikes in traffic. We therefore allow SMS traffic to use 6 of the 12 total SDCCHs, which yields a blocking probability of 1% of voice calls by the SDCCH when voice traffic requests reach $250,000$ $calls/hour$. (Note that calls would experience an average blocking probability of 71% due to a lack of TCHs with requests at this intensity.) Because these networks are designed to operate dependably during elevated traffic conditions, we believe that the above settings are realistic.

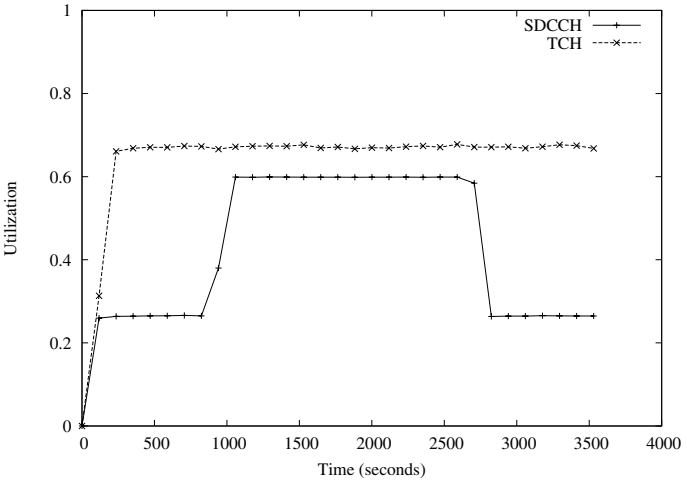

Fig. 5.18. Channel utilization under SRP

The blocking probabilities for SMS and voice flows in a sector implementing SRP are shown in Figure 5.17. Because SRP prevents text messages from competing for all possible SDCCHs, voice calls experience no blocking on the SDCCHs throughout the duration of the attack. Text messages, however, are blocked at a rate of 83%. Channel utilization, illustrated in Figure 5.18, gives additional insight into network conditions. Because calling behavior remains the same during the attack, the resources allocated by the network are more than sufficient to provide voice service to users. By design, SDCCH utilization plateaus well below full capacity. While the SDCCHs used by text messages have an average utilization of 97%, the SDCCHs used by incoming voice calls average a utilization of 6.3%. This under-use of resources represents a potential loss of utility as the majority of text messages (legitimate or otherwise) go undelivered.

The difficulty with this solution is correct parameter setting. While theoretical results indicate that allocating 10 SDCCHs only increases call blocking to 1%, voice traffic volumes fluctuate throughout the day. Provisioning resources in a static fashion must account for worst-case scenarios and therefore leads to conservative settings. While protecting the network from an attack, such a mechanism may actually hinder the efficiency of normal operation. When traffic channels are naturally saturated, as may be common during an emergency or elevated traffic scenario discussed in Section 5.6, such hard limits actually prevent users from communicating. Furthermore, as unsustained bursts of text messages are generally innocuous, such a limitation may directly impact the provider's ability to generate revenue as user perception of SMS

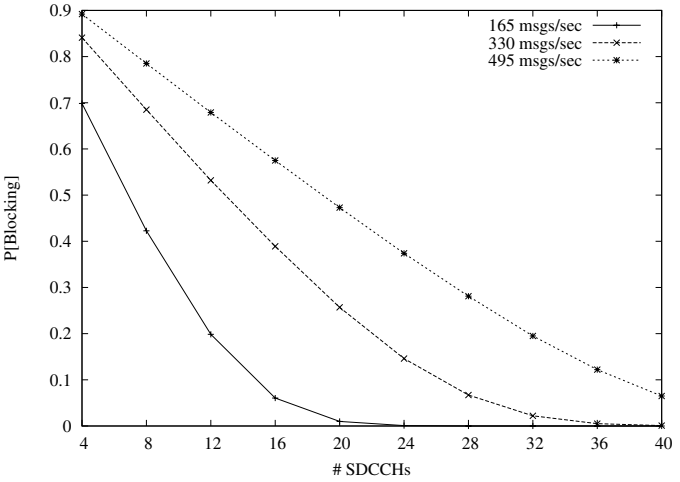

Fig. 5.19. The probability of an incoming call/message blocking in a sector for a varying number of SDCCHs.

as a real-time service erodes. Determining the correct balance between insulation from attacks and resource utilization becomes non-trivial. Accordingly, we look to our other techniques for more complete solutions.

5.10.2 Dynamic Resource Provisioning

Analysis

While SRP reprovisions capacity on existing SDCCHs, other over-provisioned resources in the sector could be manipulated to alleviate SDCCH congestion. For example, at a rate of 50,000 *calls/hour*, each sector uses an average of 67% of its TCHs. If a small number of unused TCHs could be repurposed as SDCCHs, additional bandwidth could be provided to mitigate such attacks.

Our second air interface technique, *Dynamic Resource Provisioning* (DRP), attempts to mitigate targeted text messaging attacks by temporarily reclaiming a number of TCHs (up to some limit) for use as SDCCHs. This approach is highly practical for a number of reasons. First, increasing the bandwidth (762 bits/second) of individual SDCCHs is difficult without making significant changes to either the radio encoding or the architecture of the air interface itself. Because major changes to the network are extremely expensive and typically occur over the course of many years, such fixes are not appropriate in the short term. Secondly, dynamically reclaiming channels allows the network to adjust itself to current conditions. During busy hours such as morning and evening commutes, for example, channels temporarily used as SDCCHs can

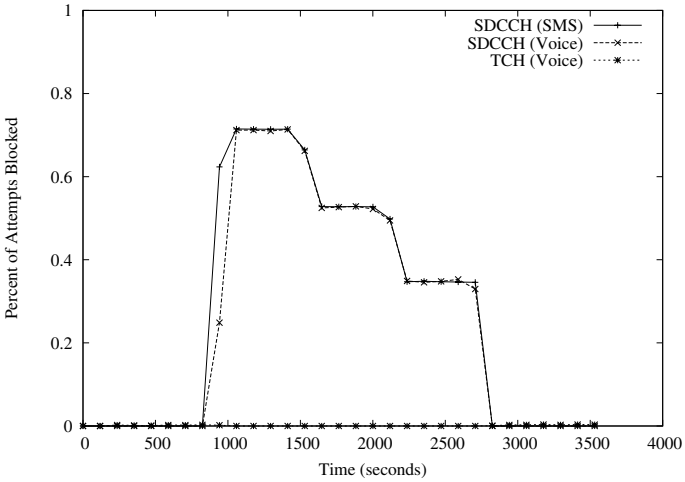

Fig. 5.20. Blocking for a sector implementing DRP

be returned to the pool of TCHs to accommodate elevated voice traffic needs. Lastly, because SDCCHs are assigned via the AGCH, allocating incoming requests to seemingly random timeslots requires almost no changes to handset software.

Figure 5.19 demonstrates the blocking probability for incoming calls and text messages in a sector using DRP to add a variable number of SDCCHs. Again, no queue was used. The ability of an attacker to block all channels is significantly reduced as the number of SDCCHs increases. Attackers are therefore forced to increase the intensity of their attack in order to maintain its potency. For attacks at a rate of 165 $msgs/sec$, doubling the number of available SDCCHs reduces the calculated blocking caused by an attack by two orders of magnitude. The blocking probability caused by attacks at higher rates, in which the number of Erlangs is greater than the number of SDCCHs, decreases in roughly a linear relationship to the number of SDCCHs added.

One potential drawback with DRP is that by subtracting TCHs from the system, it is possible to increase call blocking because of TCH exhaustion. In fact, the reclamation of TCHs for use as SDCCHs increases the blocking probability for voice calls from 0.2% in the base scenario (45 TCHs, 12 SDCCHs) to 1.5% where 40 SDCCHs are available (a reduction to 40 TCHs).

Simulation

Although it is possible to reclaim any number of TCHs for use as SDCCHs under the DRP mechanism, we limited the candidate number of channels for this conversion to two. In these experiments, a single TCH was repurposed into 8 SDCCHs every 10 minutes during the attack. This separation was designed

to allow the network to return to steady state between channel allocations. While converting only two channels is not enough to completely eliminate attacks at high intensities, our goal is to understand the behavior of this mechanism.

The blocking probabilities for SMS and voice flows in a sector implementing the DRP technique are illustrated in Figure 5.20. As TCHs are converted for use as SDCCHs, the blocking probabilities for both incoming SMS and voice requests fall from 72% to 53% and eventually 35%. This represents a total reduction of the blocking probability by approximately half. Call blocking due to TCH exhaustion was not observed despite the reduced number of available TCHs. Figure 5.21 illustrates a gradual return towards pre-attack TCH utilization levels as additional SDCCHs are allocated. The effects of the reprovisioning are also obvious for SDCCH utilization. The downward spikes represent the sudden influx of additional, temporarily unused channels. While SDCCH utilization quickly returns to nearly identical levels after each reallocation, more voice calls can be completed due to a decrease probability of the attack holding all SDCCHs at any given time.

As was a problem for SRP, determining the correct parameters for DRP is a difficult undertaking. The selection of two TCHs for conversion to SDCCHs illustrates the utility of this mechanism, but is not sufficient for real settings. To reduce the blocking probability on SDCCHs below the values observed for TCHs, a total of 48 SDCCHs would have to be made available. This leaves 39 TCHs, which results in a call blocking probability of 2.1% due to TCH exhaustion. Elevations in the volume of voice calls would likely require the release of some number of reclaimed TCHs to be repurposed to their original use.

The decision to convert channels is also non-trivial. Whereas the decision to reallocate channels at specific times was decided statically in our simulation, dynamically determining these parameters would prove significantly more challenging. Basing reclamation decisions on small observation windows, while offering greater responsiveness, may result in decreased resource use due to thrashing. If the observation window becomes too large, an attack may end before appropriate action can be taken. As was observed for SRP, the static allocation of additional SDCCHs faces similar inflexibility problems. Low resource utilization under normal operating conditions again represents a potential loss of opportunity and revenue.

5.10.3 Direct Channel Allocation

Analysis

From the security perspective, the ideal means of eliminating the competition for resources between call setup and SMS delivery would be through the separation of shared mechanisms. Specifically, delivering text messages and incoming call requests over mutually exclusive sets of channels would prevent

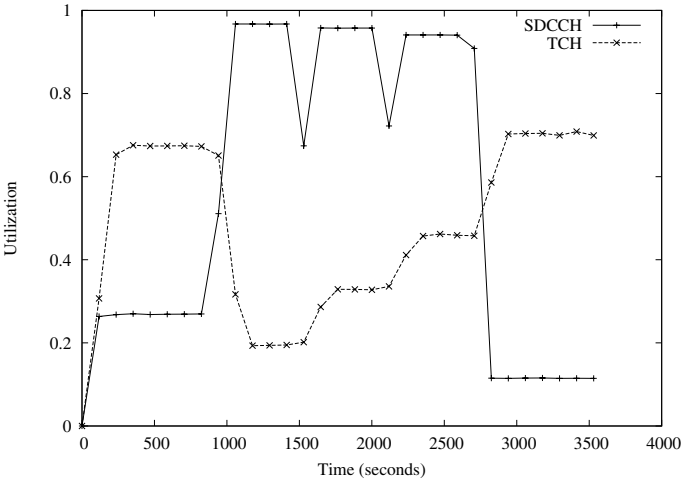

Fig. 5.21. Channel utilization under DRP

these flows from interfering with each other. The challenge of implementing such a mechanism is to do so without requiring significant restructuring of the network architecture or incurring tremendous expense. As previously mentioned, such fundamental changes in network operation are typically too expensive and time consuming to be considered in the short term. While the SRP technique provides a rudimentary separation, it is possible to further isolate these two types of traffic.

As mentioned in the previous section, DRP is easily implementable because the AGCH specifies the location of the SDCCH allocated for a specific session. After call requests finish using their assigned SDCCH, they are instructed to listen to a specific TCH. Because the use of a TCH is the eventual goal of incoming voice calls, it is therefore possible to shortcut the use of SDCCHs for call setup. Incoming calls could therefore be directed to a TCH, leaving SDCCHs exclusively for the delivery of SMS messages. This technique, which we refer to as *Direct Channel Allocation* (DCA), removes the shared SDCCH channels as the system bottleneck.

Calculating blocking probabilities for a system implementing DCA is a matter of analyzing SDCCH and TCH blocking for the two independent flows. For 165 $msgs/sec$, text messages have a calculated blocking probability of approximately 20%. This value increases to 68% as the attack intensity increases to 495 $msgs/sec$. Voice calls, at an average rate of $50,000$ $calls/hour$, have a blocking probability of 0.2%. Note that because the shared bottleneck has been removed, it becomes extremely difficult for targeted text messaging attacks to have any effect on voice communications.

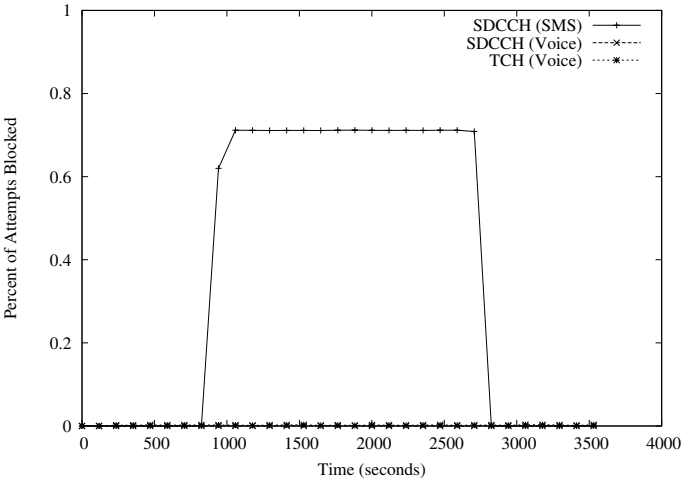

Fig. 5.22. Blocking for a sector implementing DCA

Simulation

To simulate the DCA mechanism, incoming voice calls skip directly from the RACH to the next available TCH. An average of 1.5 additional seconds was added to each incoming call duration to account for the processing formerly occurring on an SDCCH. As is shown in Figure 5.22, voice calls arriving in a sector implementing the DCA scheme experience no additional blocking during a targeted SMS attack. Figure 5.23 confirms the results in the previous figure by showing the constant TCH utilization throughout the duration of the attack. No additional assistance is provided for the delivery of text messages under DCA.

While removing the bottleneck on the shared path of SMS delivery and voice call setup, DCA potentially introduces new vulnerabilities into the network. One advantage of using SDCCHs to perform call establishment is that users are authenticated before they are assigned TCHs. Under the DCA model, however, valuable traffic channels can be occupied before users are ever authenticated. Using a single phone planted in a targeted area, an attacker could simply respond to all paging messages and then ignore all future communications from the network. Because there are legitimate reasons to wait tens of seconds for a phone to reply to a page, an attacker could force the network to open and maintain state for multiple connections that would eventually go unused. Note that because paging for individual phones occurs over multiple sectors, a single rogue phone could quickly create a black-hole effect. Such an attack is very similar to the classic SYN attack observed throughout the Inter-

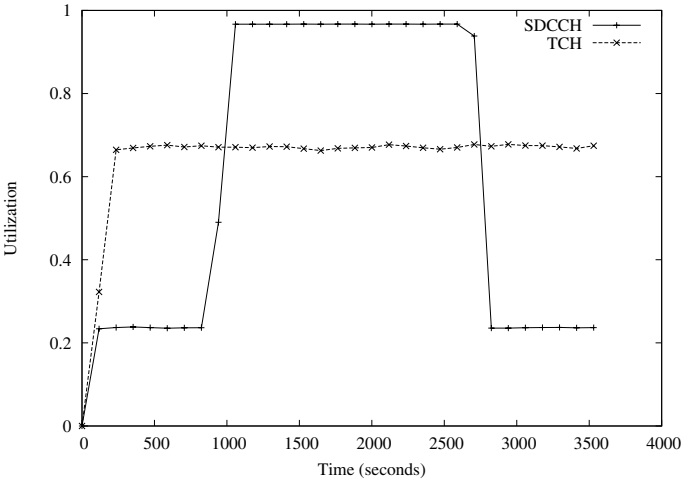

Fig. 5.23. Channel utilization under DCA

net. While seemingly the most complete, the potential for additional damage made possible because of the DCA approach should be carefully considered.

5.10.4 Summary

The resource management techniques presented above offer a number of valuable countermeasures against targeted SMS attacks. At a high level, SRP provides functionality to the weighted fair queuing approach under high load by ensuring that channels are always available to voice traffic. SRP, however, experiences even high rates of SMS blocking than weighted fair queuing (83% vs 72%). DRP allows the network to accommodate spikes in traffic by reapportioning unused TCHs, thereby making the network more flexible to a wider range of operating conditions. As we discovered in the DCA case, however, the repurposing of resources must be carefully executed so as to not introduce new vulnerabilities into the system.

5.11 Combining Mechanisms

There is no "silver-bullet" for maintaining a high quality of service for both text messaging and voice calls during a targeted SMS attack. As the above techniques demonstrate, each potential solution has its own weaknesses. The combination of such solutions, however, offers techniques robust to a wider array of threats. We examine two examples in which the fusion of mechanisms provides additional protections.

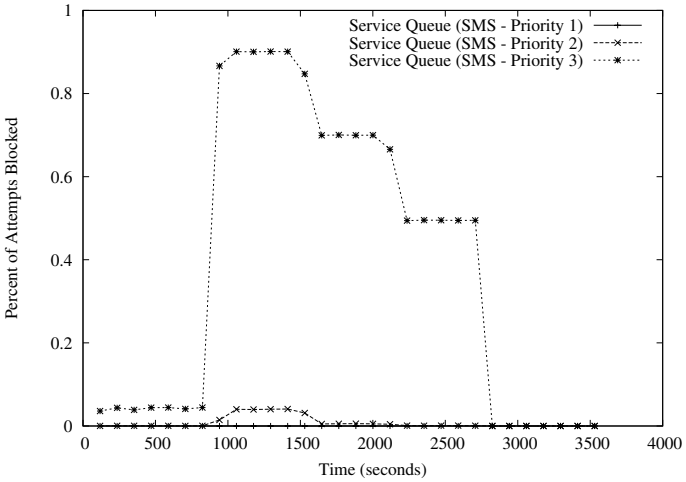

Fig. 5.24. Blocking for a sector implementing both WRED and DRP.

While directly addressing the bandwidth issue that makes targeted SMS attacks possible, the DRP technique lacks granularity to separate incoming voice and SMS requests. WRED, on the other hand, provides such traffic classification but is unable to react to attacks originating from trusted sources. To illustrate the benefits of layering these techniques, we increase the volume of legitimate traffic to 2 $msgs/sector/sec$, with 90% of that traffic being medium priority and the remaining 10% split equally between high and low priority flows. Such an increase would be representative of the elevated volumes of messages sent from crowded events such as concerts or public celebrations such as New Year's Eve gatherings. Figure 5.24 shows the result of the combination of the two techniques during an attack. Because of the naturally increased volume of legitimate traffic, subscriber-to-subscriber traffic experiences approximately 5% blocking in a sector only implementing WRED. As DRP activates and adds additional SDCCHs, only the attack traffic is dropped. Such a technique may be especially valuable during an emergency, as additional bandwidth can be provisioned to clients less likely to be malicious.

Another potentially beneficial combination is SRP and DRP. Given high volumes of voice traffic, a provider may not be able to repurpose enough SDCCHs to eliminate the effects of a targeted text messaging attack. Instead, a subset of the total channels could be reserved for voice requests. In so doing, voice blocking due to targeted text messaging attacks could be eliminated. All additional channels could be added to reduce blocking for text messages. Figure 5.25 illustrates an attack scenario in which two TCHs are reclaimed for use as SDCCHs, with 18 of 24 total SDCCHs made available to SMS. Note

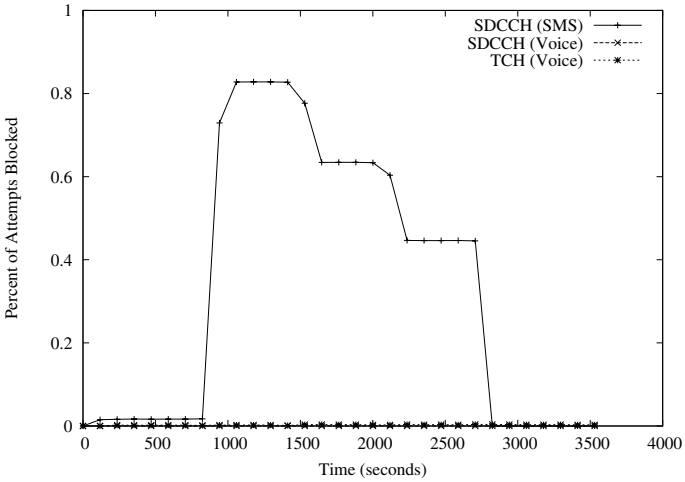

Fig. 5.25. Blocking for a sector combining DRP and SRP.

both the elimination of call blocking and the gradual reduction of blocking rates for text messages.

Other combinations are less useful. Integrating WRED and SRP, for example, would simply reduce the bandwidth made available for even high priority mechanisms. Accordingly, the network may experience decreased throughput for legitimate messages than under either scheme alone. The use of DCA with any other mechanism fails to prevent the vulnerability introduced in the previous subsection, and therefore does not warrant further investigation.

While no susceptible examples were uncovered during the course of this research, the combination of any of the above mitigation techniques should also be carefully considered. This fusion may lead to the creation of new or magnification of previously mentioned vulnerabilities. Accordingly, additional testing on development networks should be conducted before such integration could occur.

5.12 Summary

For targets ranging from individuals and metropolitan areas to entire countries, targeted text messaging attacks are capable of causing serious disruptions to service. As our modeling and simulation illustrate, an adversary with the bandwidth available to a cable modem is capable of denying service to over 70% of calls in Manhattan. We then show that currently available "edge solutions" fail to consider all possible attack vectors.

Our focus then shifted to mitigation through a number of techniques from queue management and resource provisioning. These mechanisms attempt to maintain the availability of voice telephony while providing high throughput for text messaging with varying results. WFQ and WRED offer a last line of defense by separating traffic, but fail to help the network absorb additional traffic. SRP functions similarly. The remaining two methods, DRP and DCA, allow the network to dedicate unused resources to the problem; however, DCA creates a serious new vulnerability and is therefore not a viable solution. In spite of these shortcomings, all of the above techniques except for DCA offers an effective means of mitigating the impact of targeted SMS attacks on voice calls.

The attacks discussed throughout are representative of growing and increasingly problematic class of vulnerabilities. The connectivity between the Internet and traditional voice networks introduces new avenues for exploit: once confined to exploiting only inert hosts, remote adversaries can debilitate the services we depend on to carry on our daily lives. In a broader sense, the ability to control the physical world via the Internet is inherently dangerous, and more so when the affected components are part of critical infrastructure. This work provides some preliminary solutions and analysis for these vulnerabilities. Essential future work will seek more general solutions that address these vulnerabilities in current and next generation networks.