Imperial College London
Department of Electrical and Electronic Engineering

Performance Analysis of Mobile Networks Under Signalling Storms

Mihajlo Pavloski

November 2017

Supervised by Prof. Erol Gelenbe

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy of Imperial College London

Abstract

There are numerous security challenges in cellular mobile networks, many of which originate from the Internet world. One of these challenges is to answer the problem with increasing rate of signalling messages produced by smart devices. In particular, many services in the Internet are provided through mobile applications in an unobstructed manner, such that users get an *always connected* feeling. These services, which usually come from instant messaging, advertising and social networking areas, impose significant signalling loads on mobile networks by frequent exchange of control data in the background. Such services and applications could be built intentionally or unintentionally, and result in denial of service attacks known as *signalling attacks* or *storms*. Negative consequences, among others, include degradations of mobile network's services, partial or complete network failures, increased battery consumption for infected mobile terminals.

This thesis examines the influence of signalling storms on different mobile technologies, and proposes defensive mechanisms. More specifically, using stochastic modelling techniques, this thesis first presents a model of the vulnerability in a single 3G UMTS mobile terminal, and studies the influence of the system's internal parameters on stability under a signalling storm. Further on, it presents a queueing network model of the radio access part of 3G UMTS and examines the effect of the radio resource control (RRC) inactivity timers. In presence of an attack, the proposed *dynamic* setting of the timers manage to lower the signalling load in the network and to increase the threshold above which a network failure could happen. Further on, the network model is upgraded into a more generic and detailed model, represent different generations of mobile technologies. It is than used to compare technologies with dedicated and shared organisation of resource allocation, referred to as *traditional* and *contemporary* networks, using performance metrics such as: signalling and communication delay, blocking probability, signalling load on the network's nodes, bandwidth holding time, etc. Finally, based on the carried analysis, two mechanisms are proposed for detection of storms in real time, based on counting of same-type bandwidth allocations, and usage of allocated bandwidth. The mechanisms are evaluated using discrete event simulation in 3G UMTS, and experiments are done combining the detectors with a simple attack mitigation approach.

Acknowledgements

First of all, I would like to thank my supervisor Professor Erol Gelenbe for his guidance, inspiration and patience. I am grateful that during the time under his supervision, I had the opportunity to learn from his ingenuity, logic and vivid spirit. His constructive criticism has greatly improved my scientific approach and the research presented in this thesis. I would also like to thank my examiners, Dr Toktam Mahmoodi and Professor Peter Harrison, for their through reading and helpful comments. I had the pleasure to spend my long days in the college with some inspirational colleagues, Gokce, Omer, Fred, Antoine (x2), Yuanchen, Huibo, Elif, Yasin, Lan, Yonghua, Olu, and I would like to thank them for their sincere friendship. Most of all, I would like to thank my family for their encouragement and support in my foolishness and dreams. I owe my success to them.

Contents

List of Tables

List of Figures

Preface

Statement of originality

This thesis is submitted for the degree of Doctor in Philosophy in the Department of Electrical and Electronic Engineering at Imperial College London. I certify that all material in this thesis which is not my own is acknowledged.

Copyright declaration

Publications

Parts of this thesis have been presented and published in the following peer-reviewed conferences and journals:

- Mihajlo Pavloski and Erol Gelenbe. *Mitigating for Signalling Attacks in UMTS Networks*, pages: 159-165. Springer International Publishing, Cham, 2014.

- Mihajlo Pavloski and Erol Gelenbe. Signalling attacks in mobile telephony. *11th International Conference on Security and Cryptography (SECRYPT)*, pages: 1-7, Aug 2014.

- Mihajlo Pavloski, Gokce Gorbil, and Erol Gelenbe. Counter based detection and mitigation of signalling attacks. In *12th International Joint Conference on e-Business and Telecommunications (ICETE)*, volume 04, pages: 413-418, July 2015.

- Gokce Gorbil, Omer. H. Abdelrahman, Mihajlo Pavloski, and Erol Gelenbe. Modelling and analysis of RRC-based signalling storms in 3G networks. *IEEE Transactions on Emerging Topics in Computing*, pages: 113-127, Jan 2016.

- Mihajlo Pavloski, Gokce Gorbil, and Erol Gelenbe. *Information Sciences and Systems 2015: 30th International Symposium on Computer and Information Sciences (ISCIS 2015)*, chapter Bandwidth UsageBased Detection of Signalling Attacks, pages: 105-114. Springer International Publishing, Cham, 2016.

- Mihajlo Pavloski. A performance approach to mobile security. In *IEEE 24th International Symposium on Modelling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*, pages: 325-330, Sept 2016.

Chapter 1

Introduction

Mobile communications have become inseparable part of our everyday routines, providing the means for work, entertainment, and communication. Connecting people, and businesses, it is important not only to secure the transferred data, but also to secure the stability and efficiency of the network itself. Therefore, security in mobile wireless networks, as an area of computing security, is a crucial research topic. New security issues constantly arise because of diverse reasons, such as with the unveiling of new communication technologies and services, introduction of new innovations, software and hardware updates in networking devices, etc. Another important reason is network interconnection. Mobile networks were first implemented as independent, closed and isolated systems, under full control of the Mobile Network Operators (MNOs), with tightly controlled security risks. As mobile networks evolved through time, they became more interconnected and more Internet Protocol (IP) based. With the introduction of packetised data, and the interconnection between mobile networks and the Internet, many new security challenges emerged. Connecting the two networks meant that many threats from the Internet will also become mobile network threats, and also new ones will appear [1, 2].

This thesis looks at a particular type of threat that appeared as a by-product of the interconnection between mobile networks and the Internet, and catalysed by the introduction of *smart* devices, called *signalling storms* or *signalling attacks* [3, 4, 5]. These attacks represent a Denial of Service (DoS) type of attack on the control plane of the network

and threaten the system's availability. The circumstances preceding the emergence of signalling storms were such that as volumes of mobile data increased rapidly in the past few years, MNOs concentrated on upgrading the user plane capacity, undermining the parallel increase in control data [6]. On the other side, smartphone and tablet applications tend to enable an *always-on* connectivity feeling for the users, which works with constant network re-connections and high signalling volumes [7]. Instant messaging, social networking and advertising applications, are especially problematic. Mobile networks of the second, third and even fourth generations, such as GPRS [8], UMTS [9] and LTE [10], were not designed for such pattern of data traffic, which has resulted in many service degradations and network outages [11, 12, 13, 14], and cost MNOs a total of $20 billion per year [14]. Statistics suggest that the problem will disseminate, as mobile data traffic increased by 75% in 2015 and has an estimated compound annual growth rate of 53% until 2020 [15, 6]. The share of smartphones among total devices and connections will rise from 38% in 2015 to 48% by 2020 [6]. These trends in mobile data load have proved to be proportional to the amount of generated signalling traffic [16]. On the other side, the number of security threats for mobile devices is rapidly growing [17, 18]. While from a network perspective the signalling storm would congest the signalling servers in the backbone, it would also have a negative effect for the users. An infected device would successively trigger background communication with the network and drain the battery of the device, and perhaps also create unwanted billing for accessed services [19, 20]. Moreover, with the emerging use of Machine to Machine (M2M) devices, wearable devices and the Internet of Things (IoT) concept, the problem is expected to disseminate.

This thesis analyses the performance of networks under such attacks, asking the following questions:

- Why are signalling storms happening? We identify the vulnerabilities and the circumstances in which attacks can cause problems for the network.

- What are the bottlenecks, and negative effects in the system? First, we investigate which parts of the network are mostly affected and are most likely to cause an outage. Then we look at the influence of the attacks on performance metrics of interest in

both the control and data plane, quantifying the impact from user and network perspective. The analysis is done for 3G UMTS systems, and then extended for more advanced systems beyond 3G.

- How to stop signalling storms? First, we examine if the network is capable of self-defence i.e., if it can keep its stability by adjusting some of its internal parameters. Afterwards, based on the preceding analysis, we propose mechanisms for detection and mitigation of malicious behaviour associated with these attacks.

The analysis approach in this thesis is based on mathematical modelling using stochastic techniques, and discrete-event simulation. More details on the used methodology are covered in the following Chapter.

1.1 Thesis contributions

In this thesis we presented our work on analysis of mobile networks' performance under signalling related attacks, which was carried out using techniques from mathematical modelling and discrete event simulation. Based on these analyses, we studied if a mobile network is capable of self-defence, by adjusting some of its internal parameters, and proposed mechanisms for detection of signalling attacks in real time. The main contributions of our work are listed as follows.

- We proposed mathematical models of mobile networks under signalling attack, which provided performance results from both user and network perspectives. Based on stochastic modelling techniques, such as Markov processes and queueing networks, these models could be further upgraded and adjusted to specific mobile technologies, and used for studying diverse attacks and performance metrics.
- We showed that the mobile network, in this case 3G UMTS, can adjust its inactivity timers to lower the impact of signalling attacks. Based on this, we proposed an approach with dynamic setting of the inactivity timers, as a function of the load in the network, which manages to lower the signalling load in the network, and increases

the threshold above which network failure can happen. Mitigation of signalling storms could be done by delaying the bandwidth requests of malicious users, or blocking their activity for short time periods.

- We compared mobile networks preceding, and following 3G UMTS Rel.99 which are respectively characterised with dedicated and shared allocation of bandwidth. The proposed network model included both aspects of communication: transfer of user and control data, and provided performance measures from user side and network perspectives. The numerical results showed that post-3G networks have much greater stability, and much higher signalling impacts are needed to cause service degradations, or network outages. Anyway, results also showed that looking further in the future, signalling problems in these technologies could also increase due to the rise of machine-to-machine and the internet of things trends.

- We proposed two mechanisms for detection of signalling attacks and evaluated them in a simulation environment. Both mechanisms have low computing and memory demands and would not impede the normal network operation, even if implemented on the mobile terminal side. The evaluation results show promising results: the *counter based detector*, achieved a probability of false positive detection $p_{fp} < 0.3\%$, and probability of true positive detection $p_{tp} \approx 40\%$, while the *bandwidth-usage based detector* achieved $p_{fp} \approx 0.04\%$, $p_{tp} > 95\%$, and detection delay $\tau \approx 35s$.

- As part of the more technical work in our research, we enabled an interface for remote running of simulations using Mobile Networks Security Simulator (SECSIM), described in 2.3.2. This work would contribute in the research world, as researchers interested in security of mobile networks, through the developed interface, could use the simulator at Imperial College London from anywhere in the world.

1.2 Thesis outline

The remainder of this thesis is organised as follows. In Chapter 2 we present a landscape of threats and defensive mechanisms in mobile networks and identify the positioning of

signalling attacks and storms. We also review previous work described in the literature, and introduce the scientific tools used during this research. In Chapter 3 we present two mathematical models of third generation (3G) mobile networks under signalling storm. The first model looks at the problem from the user-side, recreating the vulnerability in the mobile terminal, while the second studies the problem from network-side, analysing the radio access part of 3G. This Chapter mainly addresses the questions of why is 3G susceptible to signalling storms, and if is it capable of self-defence. Chapter 4 is concerned about upgrading the proposed network model and analysing the signalling problem in more advanced mobile technologies, such as HSPA and LTE. It further compares results for 3G UMTS and 4G LTE networks. Chapter 5 proposes two mechanisms for detection of signalling storms. Both mechanisms are implemented and evaluated in a simulation environment for UMTS networks. Finally, Chapter 6 concludes the work, outlining the contributions and possible future directions. In Appendix A we briefly describe the developed remote control of our simulator for security of mobile networks.

Chapter 2

Background

In this Chapter we first categorise and describe the most common threats in computer systems, and in particular in mobile networks, and list some of the available defensive mechanisms. We identify the positioning of the signalling storms within the threats landscape, and compare it with other types of threats. Further on, we concentrate on the signalling related attacks and review previous research work related to the field. Finally, we describe the research methodology used in this thesis.

2.1 Review of threats and defensive mechanisms

Mobile networks security represents a wide research area, and as a part of the more broad term *computing security*, is based on same principles as legacy computing systems: confidentiality, integrity and availability [21]. These three principles define the three security goals that need to be satisfied on top of *physical* security, which is also connected to the information technology, especially in near-future scenarios [22, 23]. Confidentiality, also referred to as *secrecy* or *privacy*, ensures that system-related assets are accessed only by authorised parties. Integrity stands for assets being modified only by authorised parties, while availability ensures that system assets are accessed by authorised parties at appropriate times. The opposite of availability is known as Denial of Service (DoS) [24]. Attacks on confidentiality and integrity nowadays are mainly concerned about data, both user and

control related. These kind of attacks are performed for purposes like identity theft or economic benefits, while availability attacks are focused on disruption of services or even whole systems without any direct benefits for the attacker. The positioning of this research is mainly in security of cellular mobile networks from a system design perspective in terms of availability.

A typical mobile network architecture consists of mobile terminals - also referred as User Equipment (UE), the Radio Access Network (RAN) part containing the Base Station (BS), the Core Network (CN) involving the main databases and routers, and interconnections to external networks, such as the Internet. Threats on security can break into any part of this architecture, although the most vulnerable points are the mobile terminals and the external networks, while the core network is the safest point. Mobile terminals are mostly vulnerable because of the openness of the application layer, enabling easy development of malicious applications. Moreover, the landscape of mobile devices today is very diverse, consisting of smartphones, feature phones, tablets, personal assistant devices, sensors, etc., and each of them is subject to different security threats, mostly depending of their underlying operating systems [25]. The external networks contain the user services and could be any of the following: Internet, Public Switched Telephone Networks (PSTN), corporate networks, Voice over IP (VoIP) networks, internet browsing services, interconnected mobile networks, etc. Threats originated in the external networks are also regarded as *cross infrastructure cyber attacks* [26], and can spread and mutate due to differences in organisation and functionalities. Mobile network attacks originating from the Internet are very common, knowing that mobile devices can be probed, identified and connected to, directly from the Internet [1, 2].

Table 2.1: Common security threats in computing systems.

Confidentiality / Integrity	Availability
Man-in-the-middle	Flooding
Eavesdropping	Jamming
Phishing	Tampering
Session hijacking	De-synchronisation
Alteration	Control plane flooding
Fabrication	

Table 2.2: Common defensive mechanisms in computing systems.

Defensive mechanism	Description
Human authentication	Used for both physical and computing security. • Knowledge based, eg. password, PIN, security questions • Ownership based, eg. ID card, hardware or software token • Inherence based, eg. signature, face, voice, fingerprint • Advanced biometrics, eg. retinal scan, DNA sequencing, electronic keystroke fingerprints
Data authentication and encryption	Broadly used tools, on multiple protocol layers (eg. application, network, physical layers). • Symmetric cryptography, eg. Advanced Encryption Standard (AES) (encryption only) • Asymmetric (public key) cryptography, eg. Transport Layer Security (TLS) (encryption and authentication) • Network layer tools, eg. Internet Protocol Security (IPsec) (encryption and authentication) • Other, eg. Secure Shell (SSH) (encryption and authentication)
Hosts authentication	Authentication of any interconnected device. • Symmetric and asymmetric protocols, eg. AES, TLS • Point-to-point protocols, eg. Extensible Authentication Protocol (EAP) • Network protocols, eg. Diameter
Software authentication	Assuring the software conveys its declared actions, usually performed by comparing to a database with malicious apps. • Anti-virus and anti-malware software tools • Network based solutions, eg. firewalls, gateways
Mobile network specific mechanisms	Additional layers of encryption and authentication provided by the MNO. • Additional authentication and encryption mechanisms, eg. Temporary Mobile Subscriber Identity (TMSI) • Network based solutions, eg. gateways, Network Address Translation (NAT) • Deep Packet Inspection (DPI) and honeypots [27]

In the following we will describe some of the most common threats and defensive mechanisms in computing systems, summed up in Tables 2.1 and 2.2, which are also valid for cellular mobile networks. We then focus on mobile network - specific attacks and countermeasures. Furthermore, the work in [26], [28] and [29] provides an overview and classification of the threats in mobile networks and computing systems in general, and can be used for further reading.

2.1.1 Attacks on confidentiality and integrity

In general threats on confidentiality could easily become threats on integrity of the system, which is why they are often described together, as in our case. The following threats are actually broad areas containing many sub-types of threats, and may often originate from the Internet world, but are also a threat to other networks.

- **Man-in-the-middle** - a threat that intercepts, relays and possibly alters the communication between two parties [30]. Both parties believe they are communicating directly. One example is an attack on unencrypted WiFi networks. These kind of attacks are performed because of economic and information benefits. The countermeasures are based on mutual authentication between communicating parties, such as with public key (asymmetric) cryptography where not only data is encrypted, but also the two parties are authenticated by a common trusted certification authority. Two often used application layer authentication and encryption tools are Transport Layer Security (TLS) and Secure Shell (SSH) protocols.

- **Eavesdropping** - this is generally a threat to data confidentiality, as attackers only get unauthorised access to data, but don't modify it. These attacks are common for any communication technology, intercepting data over telephone lines (known as *wiretapping*), web browsing, email, instant messaging, etc. A typical countermeasure is encryption, although attackers could also collect unencrypted metadata. A common cause of these attacks is malware, such as Trojans.

- **Phishing** - an attack acquiring sensitive information, like usernames, passwords, credit card information, etc., by an attacker pretending as a trustworthy service.

These attacks are usually spread by social media, email (as spam messages) or instant messaging. A common situation is when users enter their details in web sites with look-and-feel of popular legitimate ones. Some of the countermeasures include: social awareness, user / website authentication, email spam filtering, etc.

- **Session hijacking** - an attack in which the parameters of a connection session between two parties are stolen by a third party. This is also known as *cookie hijacking* in the Internet world, because sessions are controlled by HTTP cookies. In mobile networks, session hijacking can happen as a part of a man-in-the-middle attack on base stations. The countermeasures used include data encryption and generation of stronger session keys.

- **Alteration** - an attack with modification of data in networked devices or communication links, threatening its integrity. Most vulnerable are some IoT devices which are less secure due to their low processing and memory capabilities.

- **Fabrication** - a threat to authentication of digital systems due to attackers inserting imitation data in a normal functioning network.

2.1.2 Attacks on availability

There's a great amount of research done on attacks on system's availability, in different types of communication networks. The DoS attacks happen when one or more attackers generate flooding traffic and direct it from multiple sources towards a set of selected nodes or IP addresses in the global network. While previously these attacks were initiated by single sources, today they are typically distributed (DDoS), while the attacker uses large number of compromised devices to attack one or more devices simultaneously. The foreseen solution for this problem is to create more self-aware networks which are typically packet networks and are able to monitor their own internal behaviour, as well as the interaction with external systems, in order to modify their behaviour so as to adaptively achieve certain objectives, such as detect and react to intrusions, defend against external attacks and others [31]. The author in [32] lists the network security as one of the top priorities in future self-aware networks, knowing that the size of the Internet, and its interconnected

IP networks, are growing rapidly. Furthermore, the authors in [33, 34] propose a DoS defence system which is specifically designed for self-aware networks. The papers present experimental results that are obtained on a real networking testbed that runs the self-aware Cognitive Packet Network (CPN) routing protocol. Moreover, DoS attackers usually need to target a specific portion of mobile devices, so as to concentrate the attack on a given area, and to maximise its impact. There is also a need to search for network devices with a particular vulnerability, in such way that they can be compromised and controlled by a central point. On many occasions, such attacks have been performed using millions of IoT devices, such as IP cameras and sensing nodes. For this reason, searching for objects with particular features in large spaces is an area of interest, and is elaborated further in [35, 36, 37]. An interesting observation is the way attack detection techniques affect the search, for instance, when a deep packet inspection slows down the search, as the malignant search agents approach a given vulnerable node. Such example is presented in [38], where the authors propose a framework for DoS defence, such that islands of protection are created around critical information infrastructure. DDoS attacks specific to mobile networks are analysed in [39], where authors also point the potential bottlenecks created by the merge of GSM and IP. More extensive details are given for SMS flooding attacks, Paging attacks and attacks on dedicated channel. The authors outline the main factors to network vulnerabilities such as: network openness to Internet, deterministic system's procedures and design based on typical user behaviour. It also motivates the use of randomisation, adaptation and prioritisation as central ingredients in the design of future generation networks. The following describes the attacks on availability mentioned in Table 2.1, with more focus on attacks on mobile networks, and in particular flooding attacks which have many things in common with the signalling storms.

- **Jamming** - creation of radio interference in order to block the communication.
- **Tampering** - creation of compromised network devices, acting as legitimate hosts in the network.
- **De-synchronisation** - disruption of a connection by running incomplete procedures of the communication protocol.

- **Control plane flooding** - attacks specific to networks with a control plane, in which the flooding of the communication is performed with control messages, rather than with data packets, such as the signalling storms.

Flooding is the most common attack on availability on mobile networks. It consists a wide group of attacks characterised with repetitive patterns of communication. The goal of this attack is to exhaust its resources of the flooded system and cause interruption of its work. A few examples are:

- **Attach floods** - also known as Random Access Channel attacks. Before a mobile terminal starts communicating, it must go through an attach procedure with the network.The terminal and network authenticate each other, after which the data send over the radio interface is encrypted. This registration procedure, described in [40], includes exchange of a few unencrypted messages, which may be a vulnerability for man-in-the-middle attacks. Nevertheless, the main purpose of the attach flood is to overload the network with constant repetitions of fake attach requests initiated by the terminal. This can cause a local base station or local area network outage. If a fake International Mobile Subscriber Identity (IMSI) number is used, this attack can overload the operator's database and possibly cause a complete network outage. One variation of this attack is the *roaming attach flood* in which the attacker uses IMSI numbers of a roaming network operator and the attack is triggered by any other network partner which can increase the impact of the attack. These types of attack are easy to perform, for example any smartphone can easily be programmed to turn on and off its radio interface and re-attach to the network each time. These attacks exist since the time of GSM, and are still active. Therefore, they're easy to detect as network operators have long experience fighting against them.

- **Paging floods** - the Paging procedure takes place when a mobile terminal is inactive and has an incoming call or data and the network needs to find its location and initiate a connection. The terminal is inactive if it's in state Idle, which is an RRC state when it does not communicate for a period of few seconds up to few minutes (depending on MNO's configuration). In this state the network does not

know its accurate location. When an incoming call / data needs to be delivered, the network searches the terminal in its last reported location, called *location area*. If it's not found, the network gradually expands the search area. This procedure involves a lot of signalling, mainly in the radio access part. A paging flood attack uses this vulnerability to initiate an attack - most commonly from an external IP network, such as the Internet. The attacker needs to first discover the IP addresses of a large group of mobile terminals, which has shown to be feasible in [1, 2], and then direct some data to them. This attack can cause a complete network outage. A successful countermeasure would be to hide the terminals' IP addresses behind a Network Address Translation (NAT) mechanism, so the attackers could not find them. Although many operators don't use NATs, another concern is the use of M2M devices with public IP addresses. The use of M2M is expanding quickly, allowing devices such as sensors, tracking and identification devices to communicate with each other and control centres.

- **DNS floods** - these DoS attacks use the mobile network infrastructure - its Domain Name System (DNS) servers, to amplify the impact of attacks. The DNS services provide domain name to / from IP address translation and are one of the main components of any communication network. An LTE network for example uses three types of DNS servers: external (for hosts in external networks - such as Internet), internal (for the networks' subscribers) and intra-operator (for mobile terminals of other MNOs). The vulnerability in DNS is in its stateless procedure - a DNS server resolves its queries as they come, and cannot deny queries. This is used by attackers to send repetitive queries in which the sender's address is modified to have the address of the attacked host. When the server replies, usually with much bigger amount of information, it replies to the attacked host. If such host is an important network node, it can overload and cause full network outage. Some challenges in securing the DNS system are outlined in [41].

- **SIP floods** - SIP standing for Session Initiation Protocol, is a signalling protocol in IP networks for controlling multimedia sessions. As mobile networks tend towards all-IP implementation, the SIP protocol is used as part of the IMS system. SIP vul-

nerabilities are a potential cause for many attacks, although its risks to the network are medium. One possible attack is based on generating SIP INVITE requests at high rate, such that the SIP server or SIP Proxy would need to store a session for each request, until its resources are drained.

The DoS / availability attacks are unlike other types of attacks because the attacker can carry out a successful attack without penetrating the target network, and therefore are tough to defend against. Defence against distributed DoS attacks is even more complex, as there are multiple sources of the attack. Moreover, high intensity DoS attacks could have huge impact in short time duration. These are some of the few reasons why DoS attacks are rated as the most dangerous type of attacks to network operation. The authors in [42] compare different DoS attacks in terms of impact and risk, classifying many of them in the medium-to-high impact and risk groups. Some of the attacks described in the article include: SIP floods (medium-high risk with potential impact of outage of the voice services), DNS floods (high risk with impact of medium-full network outage), Paging floods (high risk with impact of medium-full network outage), Attach floods (medium risk with impact of medium-full network outage), etc. On the other side, there are less motivational reasons for attackers to trigger a DoS attack, compared with the confidentiality and integrity threatening attacks which could be motivated by economic benefits and information retrieval. Furthermore, the non-DoS attacks would usually target individual mobile users or network servers, rather than targeting the operation mode of the network.

Most of the attacks on system's availability have a few things in common. For example, a repetitive pattern and a requirement to have control over a large amount of mobile devices. This is not a difficult step knowing the apparent spread of malware apps. Although DoS attacks are look-alike, many times MNOs need different detection mechanisms to detect different attacks. This is because attacks target different vulnerabilities. One common countermeasure is the Deep Packet Inspection (DPI) which comes from the IP world and is widely used in the telecommunications area as well. This technique works by placing multiple inspection points through the network that are able to examine the data and metadata of traversing packets and filter out the packets pertaining possible threats to the network. This is a powerful tool, although computationally demanding,

and is used in protecting the mentioned three basic security concepts. Work in [19] uses DPI on a portion of IP packets to detect signalling attacks. The authors suggest that monitoring of the signalling data is not enough to effectively detect such attacks. Another countermeasure is known as honeypot [27]. It works by placing a *bait*, such as valuable data or service for cyber criminals, on a legitimate server or client machine and wait for it to be attacked. This type of tools are mostly used to detect malware, spam email or spam SMS. Furthermore, a network of interconnected, collaborating honeypots in known as *honeynet* or *honeyfarm*. An interesting approach of a honeypot implemented on a mobile terminal's Android system, instead on the network side, is presented in [43]. More details on the signalling-related attacks on system's availability are given in Section 2.2.

On top of data encryption in client-server communication, as an application layer countermeasure, additional security mechanisms are also employed directly by the mobile network. Firstly, in order for a mobile terminal to communicate it must go through a certain procedure with the mobile network like mutual authentication procedure and message encryption (ciphering). The authentication implementation is MNO specific although 3GPP provides some recommendations. Algorithms used in GSM, GPRS and UMTS have been broken on multiple occasions, each time triggering the need for new advancements. More detailed account on the newest security mechanisms used in LTE is presented in [44]. Apart from authentication, the network uses temporary identification numbers in order to hide important ones. In this way, the terminal is assigned a Temporary Mobile Subscriber Identity (TMSI), a frequently changed number which is used to replace the fixed International Mobile Subscriber Identity (IMSI) number for each terminal. In such way, it's harder for attackers to track or clone legitimate mobile terminals. Another countermeasure used in any IP network, including recent mobile networks, is provided by the Internet Protocol Security (IPsec). Apart from application layer mechanisms like TLS and SSH, IPsec works on the Internet (IP) layer of the protocol suite providing authentication and IP packet encryption. In order to protect information integrity, some measures like data movement monitoring and data modification logging might be implemented by some MNOs.

2.1.3 Malware

Malware is the most frequent cause of attacks on system security in the last couple of years [45]. It is a malicious software, which gets installed on user equipment using the human factor as a weak point, and can be used to disrupt the work of the network, gather sensitive information, get unauthorised access to services, and even display unwanted advertising. Since the invention of the smartphone, and other smart devices, malware infections started rapidly spreading in the mobile world as well. It is very often responsible for big network failures, such as in [11]. This is why, it is important to understand what malware is, what problems it can cause in mobile networks, and what are the available defensive mechanisms.

There have been many malware classifications in the last couple of years, like [46, 47, 48]. Anyway, the malware landscape changes very quickly, so this report uses a more top-view approach to classify malware using its goal: system damage, economic benefits and information leakage. Some common examples of *system damage malware* are:

- **Rooting** - gains control over the attacked device, also known as *jailbreak* in iOS systems;
- **Toll Fraud** - unauthorised usage of services (ex. SMS, voice calls). This malware can also gain economic benefits, for example with premium SMS sending;
- **Botnets** - enabling back-door control of devices by a remote command centre. This is one of the most common causes for DoS attacks.

Some of the malware types with a goal of *economic benefits* are:

- **Spyware** - a general term for malware that steals information in order to sell it. This type can alls be classified in the information leakage group;
- **Ransom** - steals personal information and publishes it on the internet, demanding for a ransom price to delete the information. Also belongs to the information leakage group;

- **Spam** - compromises a device in order to send out messages - very often for asking financial help. Most common examples are SMS spam and email spam;
- **Adware** - displaying unwanted advertising, which is paid-for by third parties.

And finally the *information leakage* group of malware consists the following:

- **Trojan** - a legitimate looking software, which performs malicious actions once run. This malware can also be run for economic benefits or system damage;
- **Monitoring** - a program that simply monitors the activity of the device and user and stores the information.

Malware is the source of a large portion of the attacks on mobile networks' confidentiality, integrity and availability. Current solutions require anti-malware and anti-virus software which work by checking the software's code for calls to external malicious addresses, or comparing the software as a whole to a database with previously known malicious software codes. Apart from typical anti-malware software, there are honeypot-like solutions, like [49], used for classification of Android applications. As of today, malware is the top threat to computing systems, and we can confidently claim that software authentication represents the bottleneck in today's security mechanisms.

2.2 Signalling attacks and storms

Signalling attacks and storms represent a fairly novel type of attack on mobile network's availability. In general an attack is considered a *storm* if it happens as a result of normal user activities, and *attack* if it is performed intentionally. In this thesis, both denominations are used in parallel. There have been many reported occasions where these attacks manage to degrade the quality of service, or even cause network outages in 3G UMTS networks [11, 12, 13, 14]. The cause of such incidents is well known and lies in the way we started communicating recently. Indeed, smart devices use multiple mobile apps, and each of them individually triggers communication with corresponding servers, in order to

provide their services to the customer. Most problematic are chatty Instant Messaging (IM) apps, and advertising apps. Both use frequent *background* messages either to provide an *always connected* feeling to its customers, or to satisfy its commercial partners by providing persuasive advertising [45]. All of this would have functioned perfectly well if mobile phones had PC-like Internet connectivity. On the other side, the vulnerability in today's cellular mobile networks is that the RRC mechanism is not designed for such pattern of frequent repetitive communication. Connections in mobile networks are dynamically created and torn-down to optimise the use of resources and each connection requires exchanging of multiple signalling messages to setup and reserve network resources.

In short, the vulnerability in the RRC protocol can be described as follows. In mobile networks RRC is the protocol which is responsible for management of the radio bandwidth [50]. It defines functions such as the establishment of connections and Radio Bearer (RB), mobility procedures, power control, etc. The establishment of connections and RBs is triggered by the mobile terminal, by sending a Connection Request message in UMTS or Random Access message in LTE. The RRC looks at the available resources and if possible grants them to the terminal. After the terminal has transmitted / received all data, the RRC sets up an inactivity timer after which it deducts the granted bandwidth. The length of the inactivity timers is MNO specific and depends on the RRC state of the terminal. This type of dynamic granting and deduction of bandwidth worked well in the legacy telephone systems, but experiences problems with transfer of packetised data. Signalling attacks use this vulnerability to repetitively acquire and release communication resources in order to hurt the network. Namely, each of these transitions trigger multiple signalling messages within the radio access, and even core part of the network, and if repeated by large number of mobile terminals can overload the signalling servers of the network and cause complete system failures. Section 3.1 gives more details of the RRC protocol in UMTS.

There are many possible approaches to solve the problem with excessive signalling by smart devices. In order to solve the problem on application level i.e., to encourage software developers to produce network-friendly apps, the Global System for Mobile Communications Association (GSMA) has issued a guideline for application development [51]. This guide-

line proposes a few measures including: building offline mode capabilities through caching, grouping of multiple connections together, asynchronous mode of operation, etc. At the moment, there still isn't any globally agreed solution to this problem. Most MNOs tend to run extensive offline data packet analysis to detect such malware apps, or install expensive hardware equipment in different points in their network for signalling monitoring purposes. Few large-scale, expensive, hardware and software based solutions are described in [52, 53, 54, 55]. These solutions usually work with extensive analysis of IP packets' content in many points of the network architecture. Although some of these solutions have proven partial success, network operators cannot afford to buy expensive solutions for every possible type of attack. The importance of this problem is also tackled in many international research projects, such as [56, 57]. Research done in the field focuses on quantifying the impact of such attacks on the network and on finding a simple detection mechanism. Work here can be categorised as follows:

- Problem definition and attacks classification [16, 26, 58, 39];
- Measurements in real operating networks [59], [60];
- Modelling and simulation [61, 3];
- Impact of attacks on energy consumption [20, 62];
- Attacks detection and mitigation using: counters [63, 64, 65], change-point detection techniques [66, 67], IP packet analysis [19], randomisation in RRC's functions [68], software changes in the mobile terminal [69, 49], monitoring terminal's bandwidth usage [70], detection using techniques from Artificial Intelligence (AI) [71, 72], etc.

Analyses of the system under attack are essential to understanding the cause of the problem and identifying critical system's parameters that are involved in the attack. Work done in this area represents a cornerstone for future detection and mitigation of attacks. Some of the works done in this area are based on analytical models, simulation and traffic analysis of real-world networks. The authors in [3] use a large Markov chain model for mathematical modelling of a mobile user's signalling behaviour. Its objective is to identify the system's parameters which should be avoided, namely those that, from an attacker's

perspective produce the largest amount of damage through load in the network. Results show that the load on the access part of the network increases with the increase of the attack rate, while the load on the core part of the network has a maximum value for a certain attack rate value. Furthermore, the load on the network depends on the percentage of misbehaving UEs. Authors in [61, 73] use analytical and simulation approach to analyse the effect of radio resource control signalling attacks on UMTS networks. The focal point of interest in the paper is to calculate the load on the RNC and CN in a system under attack. Additionally, the paper inspects the influence of the attack on the Quality of Service (QoS) for normal users. Results show that signalling attacks can cause significant problems in both control and user plane in the network. Furthermore, the paper makes some suggestions for lowering the impact of the attack, like enabling the cell PCH state, and provides insight how such attacks can be detected and mitigated. Some analysis of signalling traffic in real-world UMTS network is presented in [16]. The paper shows a comparison of signalling traffic by different types of mobile applications and its influence on the RRC part of the network. It also explores some application and network layer solutions for controlling application signalling traffic.

Motivated by the fact that signalling storms have repeatedly caused service degradations and outages in 3G UMTS networks, the authors in [60] conduct a set of experiments on 3G operational mobile network by modifying the software system of the UEs. The authors manage to discover RRC-related parameters set by the network, and to optimise the attack rate of a single UE such that it has the maximum impact on the network. Authors in [59] use real UMTS traffic traces and analyse them offline against different RRC state machine settings. They further compare the influence of two streaming techniques on the resource and energy consumption. The effect of the signalling storms on energy consumption and bandwidth allocation in the mobile terminals in LTE is inspected in [20, 62]. The authors show that even if small portion of the terminals in the network are misbehaving, the energy consumption of the radio subsystem of the normal UEs can increase significantly while the time spent actively communicating increases drastically for a normal data session.

Another research direction is in attacks detection and mitigation which focuses on simple user or network side solutions based on randomisation, adaptation or change-point tech-

niques. The search for a simple solution to the problem is ongoing, mostly because today there are only expensive hardware solutions on the market. One simple approach towards detection of signalling attacks in 3G networks is covered in [66]. In this publication, the authors gather signalling data by two approaches: simulation based on theoretical traffic models, and real traces from emulated 3G on a WLAN network. Attacks are detected using a statistical CUSUM method for early detection. Good points of the approach are the simplicity, dynamism and small detection time of the method, although the emulation of 3G signalling on WLAN is doubtful and some unrealistic assumptions are made. Similarly, authors in [67] work on detection of traditional flooding-based DoS attacks using a change-point detection approach with the non parametric CUSUM method. The work in [68] proposes randomisation of Radio Resource Management (RRM) and Mobility Management (MM) procedures to hide the parameters which are important to attackers. Authors try out procedures randomisation using a simple simulation of radio bearer establishment procedures. Their results suggest that the can be useful to lower the impact of attacks at the price of slight decrease in performance. The solution proposed in [19] is based on analysis of a portion of IP packets. The authors suggest that simple analysis of the signalling traffic is not enough to accurately detect malicious behaviour. Their solution is flexible in the sense that it can be installed anywhere in the data path i.e., mobile terminal, base station, gateways, etc. Results indicate that it can detect signalling attacks with more than 0.9 probability of detection and less than 0.1 probability of false alarm. The work described in [65] presents analytical analysis of a novel approach for detection and mitigation of signalling storms. The detection mechanism is based on a counter of bandwidth allocations to high speed channels, and should be implemented in each mobile terminal. When the counter reaches a certain threshold, the terminal is temporarily suspended. A more detailed description of the detector is presented in [63], while a event driven simulation approach is presented in [64]. An already functional large-scale protection system based on IP packet analysis is available in [55]. A slightly different approach towards detection/mitigation of attacks is proposed by [69] using software changes in the kernel of the mobile phone's system. The paper proposes a mobile user-side protection mechanism against signalling attacks and Trojan viruses based on partitioning the software stack into application operating system and communication partition. The communication partition

is responsible for monitoring the communication and the actual attacks mitigation. The solution is implemented in Android enabled smartphone and tried in an isolated GSM experimental network.

While most of the described work above uses tools from probability theory and queueing networks, as in the approach of this thesis, there is increasing importance of AI tools in the recent years, such as Artificial Neural Networks (ANN), which follow as an analog successor. A link between the two fields have been established in the Random Neural Network (RNN). The authors in [74] propose a generic approach to detection of DoS attacks using Bayesian classifiers combined with a Random Neural Network. The Bayesian classifiers aggregate likelihood estimation of heterogeneous statistical features, while the RNN combines them and distinguishes between normal and attack traffic during a DoS attack. The RNN is further used in [71], to detect attacks on the control plane of the network in real time, using performance metrics on the data plane. The proposed solution is suitable for implementation on the edge of the network, and does not require modification of the cellular network equipment. The data plane metrics used include: packet inter-arrival times, packet size, burst rate, destination address.

2.3 Analytical and simulation frameworks

Mathematical and simulation analysis are probably the most commonly used tools in research on security in mobile networks due to the following reasons: (i) running security threat analysis on real operating mobile network is infeasible process for protection of the sable operation of the network, (ii) MNOs usually refuse to share data due to protection of their customers and (iii) data anonymization is expensive and long process. The research methodology used in this thesis is also based on mathematical modelling using stochastic modelling techniques, and discrete-event simulation. Both approaches are similar and easily comparable. For example, both define a *system state* which can change upon appearance of a random event.

2.3.1 Stochastic modelling

The modelling approach in this thesis is based on well established stochastic modelling techniques, such as Markov processes and queueing networks [75, 76]. Markov processes are stochastic processes with a property that the next value of the process depends on the current value, but it is conditionally independent of the previous values of the stochastic process. In other words, the behaviour of the process in the future is stochastically independent of its behaviour in the past. The Markovian processes are appropriate for use in this research as they are mathematically tractable, widely used and often provide reasonable results. More specifically, we will use Poisson processes, a type of continuous time Markov processes, which have been used on multiple occasions in the telecommunication sector for traffic engineering. The Poisson processes are further suitable because of their memoryless property: the interval between two events, for example two call arrivals, is a random variable with exponential distribution with parameter λ, or equivalently mean $1/\lambda$.

Although the validity of the Poisson assumption has been questioned on multiple occasions, it has been extensively used in modelling the arrival of both voice calls and data packets in mobile networks. One such example is presented in [77] where the authors use the Poisson process to model the arrivals to a shared, time-slotted random access channel, which is commonly used in many types of wireless networks: the IEEE 802.11 standards for Local Area Networks (LANs), the Wideband Code Division Multiple Access (WCDMA) standards in 3G and the LTE standards in 4G. The Poisson processes are also used in the earliest protocols for wireless communication over shared channels, such as the ALOHA protocol [78]. The authors in [79] elaborate on the use of Poisson processes, for both transmissions and re-transmissions, in Slotted ALOHA networks. They conclude that the Poisson assumption is valid only when one allows very large packet delays compared to the slot time.

Tools from queueing theory are also used in the thesis, such as open and closed queueing networks with multiple classes of calls. The analysis of these systems is first described by Jackson [80] and Gordon and Newell [81], for finding a product-form stationary distribution

in open and closed networks with a single class of calls. Then Basket et al. [82] analyse systems with multiple classes of calls, while Gelenbe develops new product form queueing networks with negative and positive customers known as G-networks [83, 84]. While these methods suggest the use of balance equations (equating the probability flux out of a state with the probability flux into the state) in solving for the product form, more recently Harrison in [85, 86] proposed a new approach using the Reversed Compound Agent Theorem (RCAT). The RCAT methodology models the Markov processes as *dual* i.e., *reversed processes*, in which the direction of time is reversed, and manages to find new product forms for several Markovian networks [87], including G-networks [88]. In a part of this research, we look at the signalling storms blocking effect on legitimate traffic. Therefore we are particularly interested in modelling blocking networks. While usually product forms i.e., a separate solution for the network's equilibrium state probabilities, do not exist, the authors in [89] use the RCAT methodology to develop some product forms for special blocking cases, while a survey can be found in [90]. Within this research, we will use the traditional approach with balance equations for solving non-blocking systems, such as in Chapter 3. While the mentioned approaches have managed to find a product form solution for some special blocking networks, our models described in Chapters 3 and 4, will be solved numerically due to their complexity.

2.3.2 Discrete event simulation

Discrete event simulation is a widely used tool for modelling of computer networks. Unlike *continuous simulation* in which the system dynamics change continuously and are tracked over small time slices, discrete event simulation works with discrete sequence of events in time, without changes in the system state between events. It is only an event that can change the state of the system. The simulation must be able to keep track of the simulation time, and to order events chronologically in a queue. Random number generation (RNG) is an important aspect, as two simulation runs with the same initialisation (seed) of the RNGs will produce same results. Therefore, results should present an averaged outcome of multiple simulation runs with different seeds. In this thesis, we use the Mobile Networks Security Simulator (SECSIM) simulator as a discrete-event simulator specialised

in security of mobile networks, and developed within the NEMESYS project [91]. [1]

SECSIM simulator

The SECSIM simulator represents a tool for modelling, evaluation and simulation of cybersecurity in mobile networks, with a focus on the signalling layer in the radio access part. Based on Omnet++ it is an object-oriented discrete event simulator [92]. SECSIM is a modular solution, allowing network components to be easily modified using smaller components - modules. Network nodes are self-contained and independent entities, which communicate between each other via messages - modelled according to 3GPP standards for mobile protocols. Its flexible architecture enables rapid prototyping and testing of new cellular security solutions, and offers a valuable resource for evaluating different network configurations and settings. Potential users of the simulator include mobile network operators and vendors, network analysis and security companies, research institutions, and standardisation bodies. Some details on the modelled capabilities in SECSIM are summarised in the following.

- **Network entities and architectures**. SECSIM's current version contains models of functional components of UMTS networks, while components of LTE network are under construction. Some of the built components include: UE, RNC, NodeB, SGSN, GGSN, eNodeB, SGW, Internet hosts, etc. The number and size of NodeBs and eNodeBs is configurable, which enables setting up of femto, micro and macro sized cells. The RNC model has the RRC containing a single signalling server, RANAP, NBAP and GTP protocols. The signalling server plays a crucial role in the signalling attacks and their mitigation.

- **Control plane models**. In the control plane, the UE model consists of the SM, GMM and RRC layers. The networks side contains models of the corresponding entities.

[1] The Mobile Networks Security Simulator - SECSIM was built by Mr Gökçe Görbil PhD from the Intelligent Systems and Networks (ISN) group at Imperial College in London. I would like to thank Mr Görbil, and my supervisor Mr Erol Gelenbe, for giving me access to SECSIM for the research work described in this thesis.

- **Data plane models**. In the data plane, it contains the application layer containing both Circuit Switched (CS) and Packet Switched (PS) applications, the transport layer with TCP and UDP protocols and a simplified IP layer. Different types of applications that could be simulated include: web-browsing, SMS, email, multimedia streaming, VoIP calls, IM, M2M, voice calls and many types of malware apps.

- **Radio bearers**. RBs are modelled as two pairs of FIFO queues with a single server, for uplink and downlink transmission. There are signalling and data RBs. The service times of RB servers depends on the data rate associated with the RB and the length of the transmitted data.

- **Security threats**. Some of the security threats currently implemented in the simulator include, but are not limited to, the following: signalling attacks and storms, SMS spamming, premium SMS, compromised femtocells, botnets, command & control servers, etc.

- **MAC and PHY layers**. The Media Access Control Layer (MAC) and Physical Layer (PHY) are not modelled, while changes in radio conditions are modelled as random variations.

We can say that SECSIM generated data is credible, and close to real-world data, because of the following reasons: network entities of UMTS and LTE simulation models closely resemble the actual ones; communication procedures are accurately programmed by the 3GPP specifications; data generator functions on the simulated application layer are modelled by statistics of real data provided by Google [93]; propagation delays and interference on the physical layer (although not modelled directly) are represented as random variables drawn from probability distributions. Further on, SECSIM's data has been used in previous research work and published on multiple occasions [94, 61]. Our choice of SECSIM is further based on: its broad configurability options, its specialisation in the area of mobile network security, and because its code is open-source within our research group - which is an important prerequisite for testing the tools developed within this research. Nevertheless, SECSIM could still improve in areas such as: more precise modelling of the

Physical layer, enabling a finite capacity of wireless resources, organising bandwidth using spectrum reuse strategies, completion of its 4G models, etc.

2.4 Chapter summary

This Chapter first covered an overview of threats specific to mobile networks, which were categorised according the basic principles in computing security: confidentiality, integrity and availability. Most common countermeasures for each type of attack were also provided. A brief overview and classification of malware was also presented, as malware being the most prevailing threat in the mobile world in the last few years. More attention was given to attacks on network's availability, which includes the signalling related type of attacks, such as signalling storms. We reviewed research work related to the field of signalling storms and singled out the possible research directions. Finally, we described two research methodologies used in this work: stochastic modelling and discrete event simulation. The lessons learned in this Chapter are as follows:

- Looking at the broad field of computing security, we can conclude that the currently available defensive mechanisms for authentication and encryption would work well if implemented properly. Anyway, most of the security breaches today happen because of human error, incompliance with security standards, and poor system design.
- Among the defensive mechanisms in computing security (human, data, host, and software authentication, and data encryption), the software authentication field currently represents the bottleneck. Therefore, malware is the primary source of security problems in today's information systems.
- Most of the availability attacks in mobile networks, and in the global Internet, follow a common repetitive pattern, which also plays a key in their detection.
- The mobile network attacks on availability would usually need to have compromised a huge number of mobile devices in order to make an impact on the network. Anyway, with the invention of smart devices this task for the attackers is simplified because devices could be compromised using malware.

- Most of the vulnerabilities for DoS attacks in mobile networks happen because of poor protocol design or simply because of protocols being outdated. For example, the RRC protocol in UMTS was initially designed for voice calls as the primary type of traffic, but packetised traffic has quickly taken over, which initiated several problems for UMTS.

- The signalling attacks/storms could happen because of intentional malicious activity, or as a by-product of a legitimate network functions, for example a signalling-demanding mobile application becoming too popular or mobile devices re-trying to connect to a failed server.

- Most of the research in the field of signalling storms works on their detection. Some of the approaches include: traffic monitoring with change-point algorithms, randomisation of protocol messages, counting repetitive actions, deep packet inspections, algorithms from anomaly detection, etc.

- There have been very few research publications where analysis are done on data of real working mobile networks. In the case where real-world data is not available, the usual approach, as in this research work, is by using mathematical modelling and simulation tools.

In the next Chapter, we start by describing the Radio Resource Control vulnerability in UMTS, and then propose two stochastic models (respectively looking from the terminal and network perspectives) in order to understand the signalling problem better, and find answers to our first research questions.

Chapter 3

Signalling storms in 3G mobile networks

Mathematical modelling is an important tool used in research that provides the means for analysis and synthesis of systems. It is especially important in cases with deficiency of real-world data. This approach is also suitable in the scope of our work, because it could represent a smaller part of the system, and focus on the vulnerability, isolated from the other conditions in the system. As seen in Chapter 2, the problem with signalling related attacks on networks's availability lies in the design of communication protocols and is caused by recent technological advances, such as the invention of smart devices. There are negative consequences for both the mobile user, and the mobile network operator, and the problem could be tackled from both perspectives. This Chapter looks at the signalling storms problem in 3G UMTS from both perspectives using stochastic modelling techniques, described in Section 2.3. As a starting point, the 3G UMTS system is selected because of its dominant implementation worldwide, and because of the many documented signalling problems in the system. The RRC state machine is first described in Section 3.1 identifying the vulnerability of interest. Afterwards, Section 3.2 presents a model of the vulnerability on the terminal side, while Section 3.3 presents the network side model. Our conclusions are listed in Section 3.4.

3.1 Radio Resource Control vulnerability

The management of communication resources in UMTS is regulated by the RRC mechanism. In general, there are two RRC connectivity modes: Idle and Connected. In Idle mode there aren't any radio resources used between the mobile terminal, referred as User Equipment (UE) in UMTS, and the Radio Network Controller (RNC). The few tasks a UE performs in Idle mode are related to neighbour cell monitoring, cell re-selection, paging and reception of broadcast data. In this state, the UE consumes the least amount of energy. In Connected mode, there is a logical connection established between the UE and the network, although physical communication channels may or may not be allocated. RRC's Connected mode is further divided in four states:

- CELL_DCH (in short *DCH state*) - a state where a dedicated connection exists in uplink (UL) and downlink (DL) direction. Radio resources are dedicated exclusively to the UE allowing it to send and receive data at high rates up to around 10 Mbps. In this state the UE consumes the highest amount of energy;
- CELL_FACH (*FACH state*) - there aren't any dedicated connections but data can be transferred via common channels. This state is suitable for transfer of small amount or bursty data. The data rate achievable is up to around 10 kbps and there's moderate battery consumption;
- CELL_PCH (*PCH state*) - similarly to Idle state the UE monitors only the paging and broadcast channels. The difference is that the logical RRC connection still exists;
- URA_PCH (*URA state*) - a state similar to CELL_PCH where every cell change does not trigger a cell update procedure in order to decrease the signalling activity.

In UMTS there are two concepts for data communication: the concept of *connection*, and the concept of Radio Bearer (RB). When an idle UE wants to make a data call it needs to establish a connection and obtain communication resources. The UE first initiates establishment of a RRC connection and then the network creates one or more

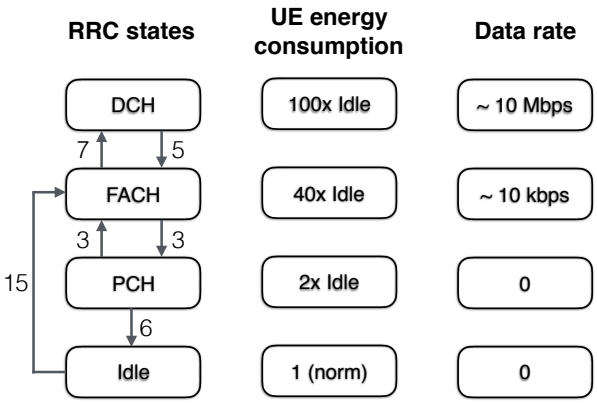


Figure 3.1: RRC states in UMTS. Typical number of generated signalling messages for state changes (left), energy consumption (middle) and the maximum data rate for the mobile terminal (right).

RBs depending on the requested and available resources. There can be only one RRC connection per data call or per UE, but many RBs within one connection. The RB defines the properties of the connection depending on the requested QoS parameters. For instance, to transfer low volume data the UE will obtain a common physical channel (FACH state) and a dedicated physical channel (DCH state) for a higher volume, delay-restricted data. After data is transmitted, the network then revokes allocated resources after an inactivity timeout t_L in FACH state or t_H in DCH state which are in order of few seconds [95]. The vulnerability in the RRC state machine is that moving between these states generates multiple signalling messages in the radio access and core parts of the network. Moreover, the frequency of moving is controlled by simple timers defined for each state. Fig. 3.1 shows the four basic RRC states, omitting the URA state, the typical number of signalling messages generated with each state switch, the approximate energy consumption of the UE and the maximum data rate achievable.

A UE can send and receive data using either a forward access channel (in FACH state) or a dedicated access channel (in DCH state) depending on the amount and type of data it has to send. An attacker could do the same and successively request either a FACH or DCH channel, therefore we say it performs a *FACH attack* or *DCH attack*, respectively. In order to maximise the number of connections per unit time, in a FACH attack, the usual

state change is between PCH and FACH states, while in a DCH attack - between FACH and DCH states.

3.2 Mobile terminal model

This Section models the system's vulnerability on the mobile terminal side, in order to analyse the problem at the root of its cause - at the point where signalling is generated. The goal is to see if the network/system can maintain its stability under an attack by changing some specific state transition time constants. More precisely we are interested if the inactivity timers and state transition delays, which are fixed parameters in current network setups, are able to be configured in a certain way so the impact of the attack is reduced or completely evaded. The Section first describes the mathematical model of the RRC mechanism under attack, then defines a cost function regarding the *normal* and *attacked* states of the model which is minimised numerically. Finally results show the value of the cost function regarding the parameters of interest.

3.2.1 Model description

The modelling approach taken in this Chapter comes from the field of stochastic / random processes. More precisely, we are using Markov processes with a particular set of states - the values that the system can take. One property of the Markov processes is that the next state that the system can take depends only on the current state, but not on previous states. The system we are modelling in this Chapter is a single user's RRC part of UMTS under signalling attack and is described by the state diagram on Fig. 3.2. The figure depicts a model derived from the conventional RRC state machine in UMTS with added *attack* states in the system. The idle state is represented by D - Dormant, PCH and URA states are represented by a single P state, L (the low state) represents FACH and H (the high state) represents the DCH state. The corresponding attack states are indicated with L_A and H_A for attacks on the respective low and high states.

At any given time and state, the system may receive one of the following four requests

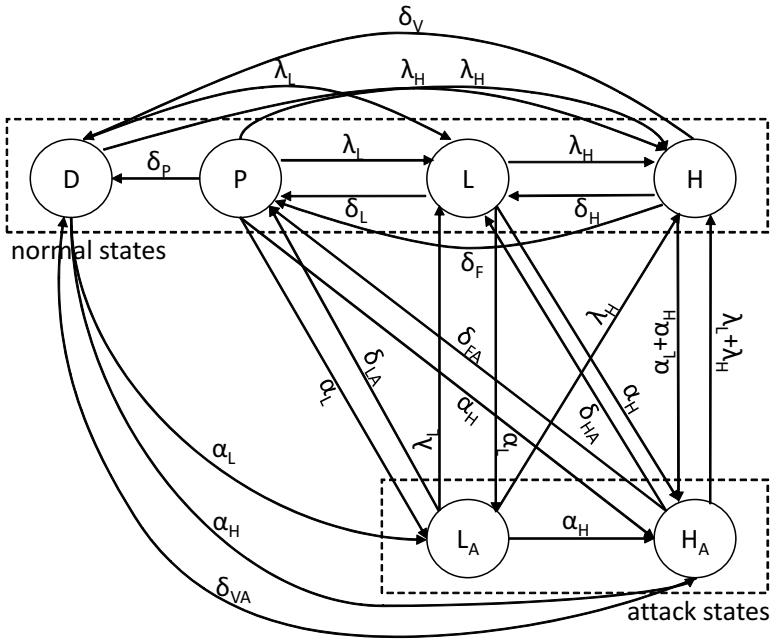


Figure 3.2: RRC state machine model of UMTS under signalling attack.

triggered by the UE: normal FACH, normal DCH, attack FACH and attack DCH request which trigger the promotion transitions in the system. Let us denote with λ_i the rate of normal requests for state i and with α_i the rate of attack ones where $i \in \{L, H\}$. We define the *attack ratio* parameter k as:

$$k = \frac{\alpha_L}{\lambda_L} = \frac{\alpha_H}{\lambda_H}. \tag{3.1}$$

State demotion rates from normal states are denoted by $\delta_P = \frac{1}{t_P}$, $\delta_L = \frac{1}{t_{FACH}+t_L}$ and $\delta_H = \delta_F = \delta_V = \frac{1}{t_{DCH}+t_H}$, where t_{FACH} and t_{DCH} represent the average duration of data transmission in the respective states while t_x is the inactivity timeout period in state x. Transitions denoted by δ_F and δ_V represent the *fast dormancy* mechanisms which were introduced in later versions of UMTS standards. During a signalling attack, the attacker usually does not transmit any data because the purpose of the attack is solely to trigger the signalling transitions. Therefore, the demotion rates from the attack states are selected as $\delta_{LA} = \frac{1}{t_L}$ and $\delta_{HA} = \delta_{FA} = \delta_{VA} = \frac{1}{t_H}$. Two specific cases are included when low-bandwidth (FACH) requests are served in dedicated channel states, represented by the transitions from H to H_A and vice-versa.

To analyse the system in steady state, one needs to find the equilibrium distributions i.e., the state probabilities. Here, we denote the *probability of state i* with π_i. Equating the probability flux out of a state to the probability flux into the state, we get the following balance equations:

$$\begin{aligned}
\pi_D(\lambda_L + \lambda_H + \alpha_L + \alpha_H) &= \pi_P\delta_P + \pi_H\delta_V + \pi_{HA}\delta_{VA},\\
\pi_P(\lambda_L + \lambda_H + \alpha_L + \alpha_H + \delta_P) &= \pi_L\delta_L + \pi_H\delta_F + \pi_{LA}\delta_{LA} + \pi_{HA}\delta_{FA},\\
\pi_L(\lambda_H + \alpha_L + \alpha_H + \delta_L) &= (\pi_D + \pi_P + \pi_{LA})\lambda_L + \pi_H\delta_H + \pi_{HA}\delta_{HA},\\
\pi_H(\delta_V + \delta_H + \alpha_L + \alpha_H + \delta_F) &= (\pi_D + \pi_P + \pi_L + \pi_{LA})\lambda_H + \pi_{HA}(\lambda_H + \lambda_L),\\
\pi_{LA}(\lambda_L + \lambda_H + \alpha_H + \delta_{LA}) &= (\pi_D + \pi_P + \pi_L)\alpha_L,\\
\pi_{HA}(\lambda_L + \lambda_H + \delta_{HA} + \delta_{FA} + \delta_{VA}) &= (\pi_D + \pi_P + \pi_L + \pi_{LA})\alpha_H + \pi_H(\alpha_L + \alpha_H).
\end{aligned} \tag{3.2}$$

The system of linear equations could be solved by taking into account the normalisation condition $\sum_i \pi_i = 1$, which would give a solution for the equilibrium distributions. Our optimisation goal is to minimise the time spent in the attack states i.e., to minimise π_{LA} and π_{HA}, thus maximising the normal behaviour of the system. To optimise the system, we can define a cost function C as follows:

$$C = \frac{\pi_{LA} + \pi_{HA}}{\pi_L + \pi_H}. \tag{3.3}$$

The cost function can be considered as a function of two variables: the inactivity timers t_L and t_H, and *call setup delays* t_{xL} and t_{xH} in promotion transitions to FACH and DCH. From an implementation perspective, the variables t_L and t_H are set on the network side by the MNO, while t_{xL} and t_{xH} are delays that should be implemented on user side, before any signalling is triggered. While modifying the inactivity timers is fairly straight-forward, inserting delay in promotion transitions should be looked at from the system's perspective. Let us denote with θ_i the *total request rate for state i* seen by the system:

$$\theta_i = \lambda_i + \alpha_i, \quad i \in \{L, H\}. \tag{3.4}$$

Then the *average inter-request interval* is:

$$t_{\theta_i} = \frac{1}{\theta_i}. \tag{3.5}$$

We insert a setup delay t_{xi} in transitions to state i and get the *delayed inter-request interval*:

$$t'_{\theta_i} = t_{\theta_i} + t_{xi}. \tag{3.6}$$

Solving for the new - *delayed arrival rates* we get:

$$\lambda'_i = \frac{\lambda_i}{1 + t_{xi}\lambda_i(k+1)}, \quad \alpha'_i = \frac{k\alpha_i}{k + t_{xi}\alpha_i(k+1)}. \tag{3.7}$$

which represent the *delayed* normal and attack rates at state i. In order to minimise C we can use the partial derivative of C with respect to both variables: $\frac{\partial C(t_i,t_{xi})}{\partial t_i}$ and $\frac{\partial C(t_i,t_{xi})}{\partial t_{xi}}$, and a function minimisation algorithm, such as gradient descent. The following Section summaries the numerical results for the system.

3.2.2 Numerical results

The results presented in this Section inspect the value of the cost function $C(t_i, t_{xi})$ depending on the two candidate defensive mechanisms, inactivity timers and call setup delays, in case of an attack on FACH or DCH state. In the analysis, we use the following values for the system's parameters:

- We can calculate the duration of data transmission in FACH and DCH channels, t_{FACH} and t_{DCH}, as the quotient of the data volume per channel allocation and the channel data rate. For this purpose, we can take the maximum data rates for a Rel 99 UMTS version of FACH and DCH channels as 32 kbps and 2 Mbps, respectively [96, 97, 98]. The average data volume per background (FACH) allocation is 100 KB [99], and 320 KB per high-speed (DCH) allocation as an average web page size [93]. Using these values, we get $t_{FACH} = 3.125$s and $t_{DCH} = 1.28$s.

- We select some common values used for the inactivity timers in UMTS, in order of

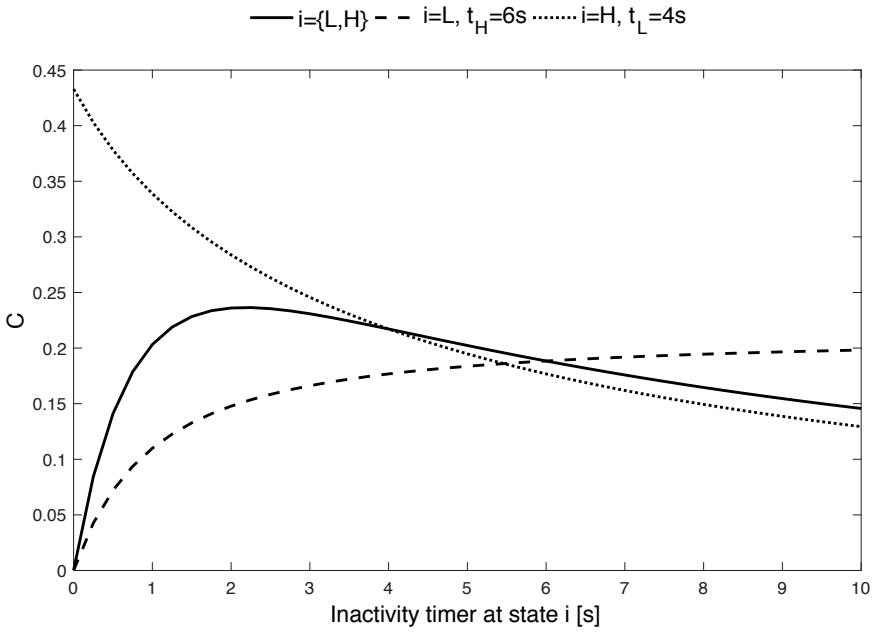


Figure 3.3: The cost C as a function of inactivity timeout period at state i for FACH attacks.

few seconds fro FACH and DCH state timers, $t_L = 4$s and $t_H = 6$s, and in the order of minutes for the PCH state timer, $t_P = 20$min [4].

- The value of the attack ratio k is only a measure of attack strength, and as so it only influences the amplitude of C.

First, we look at attack on FACH state and as a defensive mechanism we modify the two inactivity timers in the system. Three scenarios are considered: (1) we use a fixed value for $t_L = 4s$ and modify t_H, (2) we use a fixed value for $t_H = 6s$ and modify t_L, and (3) we modify both timers together using $t_L = t_H$. Results are presented on Fig. 3.3. For fixed $t_L = 4s$ the cost function decreases with the increase of t_H. This simply shows that the longer the system stays in H state the lower the impact of the attack on FACH state. For fixed $t_H = 6s$ the cost function increases with the increase of t_L meaning that the quicker the system returns to normal state, the lower impact of attack. The cost function for changing both t_L and t_H together has a more complex form, rising to a certain point after which it starts declining. Of course, the cost function has a minimum at $t_L = t_H = 0$ i.e., when timers are turned off. Anyway, very low timer values have shown to be unsuitable in practise as it would trigger higher number of timeouts and re-connections. Therefore a

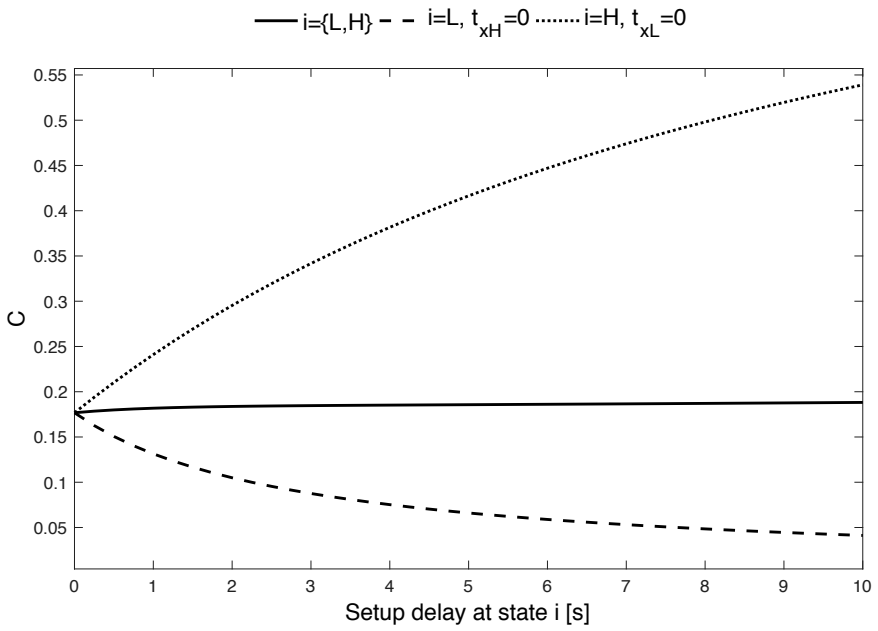


Figure 3.4: The cost C as a function of setup delay at state i for FACH attacks.

better choice is selecting higher values for the two timers.

On Fig. 3.4 again we look at an attack on FACH state, but now modify the call setup delays t_{xL} and t_{xH}, while inactivity timers are kept to their default values. Three scenarios, analog to the previous figure are inspected: modifying the two parameters together, and modifying one parameter while the other is fixed to 0s. Results show that setting $t_{xH} = 0$ and increasing t_{xL} is a good choice for lowering the attack. In contrast to that, increasing the delay of DCH requests while an attack is ongoing on FACH state sharply increases the impact of the attack. Increasing the delay in both FACH and DCH requests at the same time does not introduce any improvements.

In the following two figures, we will look at a system under DCH attack. Fig. 3.5 shows the results for using the inactivity timers as a defensive mechanism, while keeping the setup delays fixed to their default values. Results are analogous to the ones in Fig. 3.3. Increasing the inactivity timeout in the un-attacked state t_L introduces small improvements, while in the two other scenarios the cost function has convex form which suggests that higher values for the timers are more suitable. There is one difference to the case in Fig. 3.3: although we would expect constant increase in C with the increase of t_H when t_L

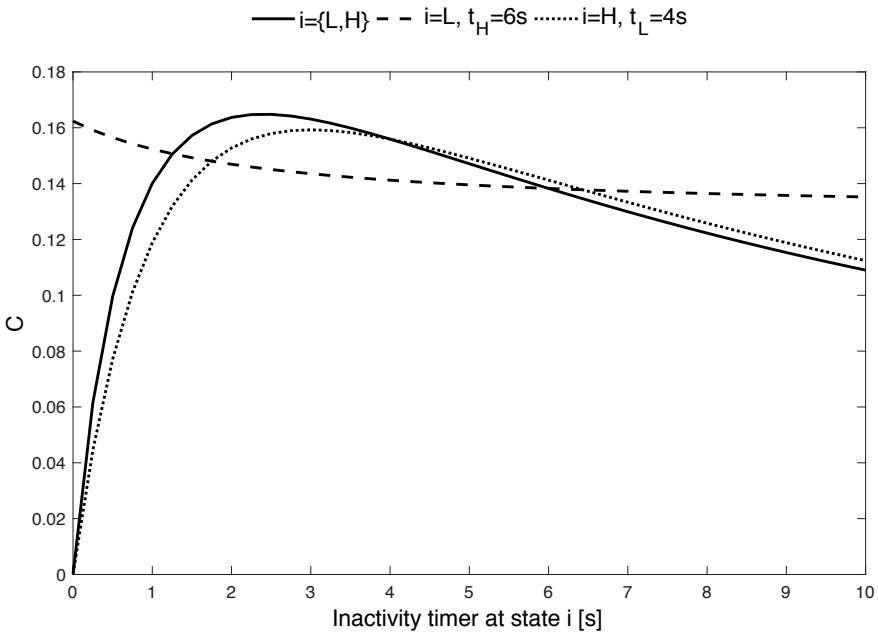


Figure 3.5: The cost C as a function of inactivity timeout period at state i for DCH attacks.

is fixed, C drops after a certain point. This is due to normal FACH requests being served in high bandwidth channels DCH, thus the transition H_A to H being more probable than H to H_A.

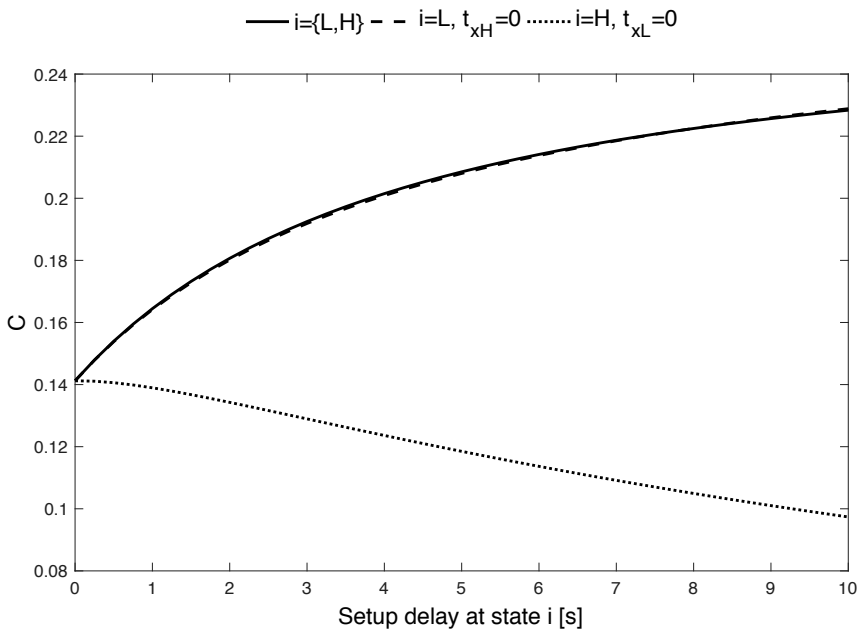


Figure 3.6: The cost C as a function of setup delay at state i for DCH attacks.

Finally, Fig. 3.6 depicts the case of inserting delay in promotion transitions in a system under DCH attack. Analog to the case of FACH attack, inserting delay in DCH requests in this case lowers the cost function, while inserting delay in FACH requests increases it. C also increases if increasing the setup delay for FACH and DCH requests at the same time.

In general, the cost function has a minimum for FACH attack and $t_L = 0$, or DCH attack and $t_H = 0$. This result is correct by means of lowering the cost function. Anyway selecting small timers in both cases means higher number of timeouts i.e., higher number of transitions. In case of FACH attack, setting the inactivity timer of DCH state to higher values is a good choice. Similarly, selecting higher values for the inactivity timer in FACH state slightly improves the security of the system under DCH attack. When adding setup delay to connections, the conclusion is that delay has to be added only to transitions towards the attacked state. If the attacked state is not correctly estimated, adding delay to the wrong transitions would have a negative effect.

3.3 Mobile network model

This Section presents a mathematical model, based on techniques in queueing theory, of the radio access part of 3G UMTS network under signalling storm. The goal of the modelling in this Section is: (i) to examine the influence of the attack on the network and to identify the points with highest impact, and (ii) similarly to Section 3.2, to see to what extent some system's parameters can be used in the defence against such attacks, in particular - the inactivity timers in RRC. This part first describes the general model of the network, and afterwards focuses in more details on the models of data channels and the models of normal and malicious user behaviour. Finally a dynamic timer is proposed and compared with a default static one.

3.3.1 Model description

The proposed network model is depicted on Fig. 3.7 and focuses on the radio access part of a typical mobile network with N cells and a single radio network controller. It represents an open network model with calls joining it with a Poisson arrival rate λ_0^r. The arrival process does not depend on the state of the system. There are two classes of *calls* traversing the network, with classes denoted with $r \in \{n, a\}$ for *normal* (traffic for SMS, web-browsing, instant messaging, etc.) and *attack* (malware-generated malicious traffic), respectively. There are two types of service centres (nodes) in the model. First, the Signalling Server (SS) in the network controller is modelled as a first-come-first-served (FCFS) single queue with infinite waiting places and service rate μ_s. Second, the data channels in each cell i are modelled with an $M/M/m_i/m_i$ queueing model as m_i parallel servers without queueing option. The finite number of servers is due to the limitation of frequency bands allocated to MNOs. The service times in each node in the network are exponentially distributed. In the data channel nodes, the service time distribution is distinct for different classes of calls. This is because of the different bandwidth usage behaviour of the normal and malicious calls and its model is described in Section 3.3.3. Furthermore, the service time in the data channel node i is state dependent i.e., depends on the number of calls in service n_i. The service time distribution for the SS node is same for both classes of calls, because the signalling procedure undertaken by the network does not distinguish call classes. To simplify the analysis we assume that all rates within the network are Poisson, although this is not the case with nodes with finite capacity and possibility of blocking. The rest of the notations used in the model are listed in the following:

- λ_s^r - the total rate of class r calls coming to the signalling server. It includes the calls joining from outside the network, the calls that have successfully been served and return as new calls, and the calls that retry transmitting/receiving data after not being allowed due to insufficient free channels;
- p_{bi} - the probability of a call at cell i is not admitted i.e., is blocked, for communication because all data channels are occupied. This quantity is same for both types

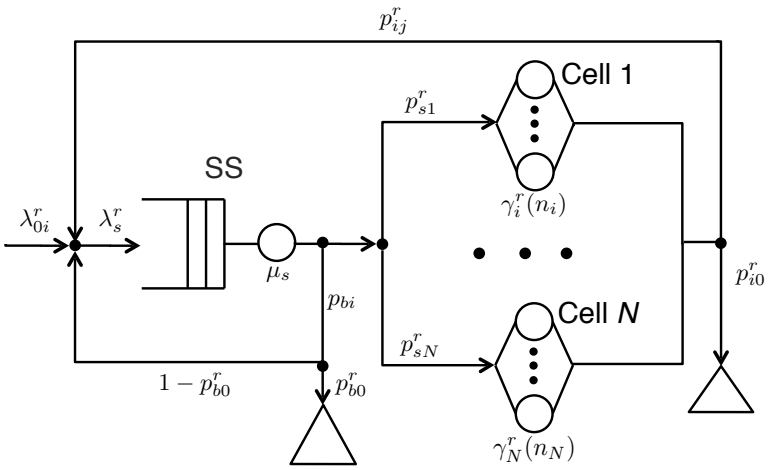


Figure 3.7: Queueing model of the radio access part of a mobile network.

of call classes;

- p_{bo}^r - the probability that a blocked call of class r leaves the network. p_{b0}^a actually gives the stubbornness of the attacker with values close to 0 denoting an attacker who triggers repetitive calls although not awarded any data channels for communication. In a similar way, p_{b0}^n is a measure for human persistence;

- p_{si}^r - the probability that a class r call which has finished its signalling procedure, will join data channels in cell i;

- $\gamma_i^r(n_i)$ - the service rate of class r calls at cell i, dependent on the current number of calls in service n_i;

- p_{i0}^r - the probability that a class r call leaves the network after successful service;

- p_{ij}^r - the probability that a class r call joins cell j after successful service in cell i.

Note that the superscript in some of the above notations is an indicator of the corresponding class, and not a symbol for exponentiation. Solving the balance equations for our model would be a complex task given the different class service distributions and FCFS queueing approach. Therefore, in the rest of this Section we look at numerical analysis of the proposed model. To solve the system numerically and simplify the exposition, we will drop the subscript i referring to the cell number. The arrival rate at the signalling server

of class r calls is given by:

$$\lambda_s^r = \frac{\lambda_0^r + \gamma^r \cdot (1 - p_0^r)}{1 - p_b \cdot (1 - p_{b0}^r)}, \tag{3.8}$$

while the total arrival rate is a sum of the arrival rates of both classes of calls: $\lambda_s = \lambda_s^n + \lambda_s^a$. To solve this equation, we need to find γ^r and p_b, which is done in the following.

3.3.2 Data channels model

We can describe the steady state of the system by the pair (n^n, n^a) where n^n and n^a are the number of normal and attack calls in service. If we denote with μ^r the service rate of a single data channel for class r call we get a two-dimensional state diagram as on Fig. 3.8. Note that the total arrival rate of class r calls requesting a free communication channel is given with λ_s^r. The cell could have at most m calls in service and so has finite number of states represented by the M/M/m/m Markov chain model.

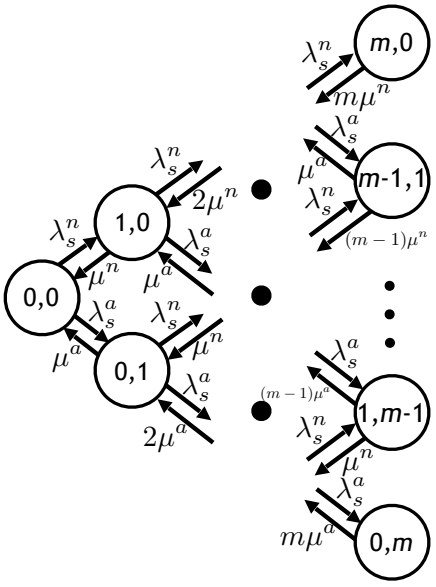


Figure 3.8: State diagram for of the data channels model with m parallel channels.

Lets denote with π_{ij} the probability of i normal and j attack calls in the node, we can

write the following balance equations:

$$\begin{aligned} \pi_{00}\lambda_s^n &= \pi_{10}\mu^n, \\ \pi_{00}\lambda_s^a &= \pi_{01}\mu^a, \\ \pi_{10}\lambda_s^n &= \pi_{20}\cdot 2\mu^n, \\ &\dots \end{aligned} \tag{3.9}$$

Solving each equation for π_{00} and propagating the result as $i, j \to m$, for π_{ij} we get:

$$\pi_{ij} = \frac{(\rho^n)^i(\rho^a)^j}{i!j!}\pi_{00}, \quad 0 < i + j \le m \tag{3.10}$$

where $\rho^n = \frac{\lambda_s^n}{\mu^n}$ and $\rho^a = \frac{\lambda_s^a}{\mu^a}$ are the *utilisations of a cell due to normal and attack calls.* Using the normalisation condition $\sum\limits_{\forall i,j; i+j\le m} \pi_{ij} = 1$, for π_{00} we get:

$$\pi_{00} = \frac{1}{\sum\limits_{j=0}^{m}\sum\limits_{i=0}^{m-j}\frac{(\rho^n)^i(\rho^a)^j}{i!j!}}. \tag{3.11}$$

We can further define the *marginal probabilities* for a fixed number i of normal or attack calls in the node as:

$$\pi^n(i) = \sum_{j=0}^{m-i}\pi_{ij}, \quad \pi^a(i) = \sum_{j=0}^{m-i}\pi_{ji}. \tag{3.12}$$

Finally we can calculate the *average number of normal and attack calls* $\bar{n^n}$ and $\bar{n^a}$ in the node:

$$\begin{aligned} \bar{n^n} &= \sum_{j=1}^{m} j\pi^n(j) = \sum_{j=1}^{m}\sum_{i=0}^{m-j}\frac{(\rho^n)^j(\rho^a)^i}{i!j!}j\pi_{00}, \\ \bar{n^a} &= \sum_{j=1}^{m} j\pi^a(j) = \sum_{j=1}^{m}\sum_{i=0}^{m-j}\frac{(\rho^n)^i(\rho^a)^j}{i!j!}j\pi_{00}. \end{aligned} \tag{3.13}$$

And for the total *average number of calls in the node*, regardless of the call type, we get:

$$\bar{n} = \sum_{j=0}^{m}\sum_{i=0}^{m-j}(i+j)\pi_{ij} = \sum_{j=1}^{m}\sum_{i=0}^{m-j}\frac{(\rho^n)^i(\rho^a)^j}{i!j!}(i+j)\pi_{00}. \tag{3.14}$$

Here $\bar{n} = \bar{n^n} + \bar{n^a}$ is also valid. As mentioned earlier, the call blocking happens regardless of the call class, so the *probability of blocking* p_b can be calculated as the probability of m total number of calls in the node:

$$p_b = \sum_{i=0}^{m} \pi_{i(m-i)} = \sum_{i=0}^{m} \frac{(\rho^n)^i (\rho^a)^{m-i}}{i!(m-i)!} \pi_{00}. \tag{3.15}$$

The solution for the blocking probability p_b, average number of calls in the system $\bar{n}$, and the rate λ_s^r is found by solving the system of non-linear equations.

3.3.3 User behaviour model

In general, the two classes of calls, referred to as *normal* and *malicious* (or *attacking*) have different service time distributions. A normal call, for example a session of web browsing traffic, would usually happen in bursts which would occupy the channel for a longer period. Contrary, attack calls would usually transfer only a small portion of data in order to trigger quick bandwidth allocations and deallocations, as previous research has identified [61]. The two patterns are depicted on Fig. 3.9 with T^n denoting the *normal session duration* and T^a the *attack session duration*. In this context, the *inactivity timeout period*, denoted with t_0 plays a crucial role. Using this timeout MNOs can control the dynamic allocation and reuse of bandwidth and is usually a few seconds long. The selection of t_0 represents a trade-off between the amount of used bandwidth and the amount of signalling messages due to new allocations. The symbols s and q on Fig. 3.9 stand for *service* and *quiet* periods.

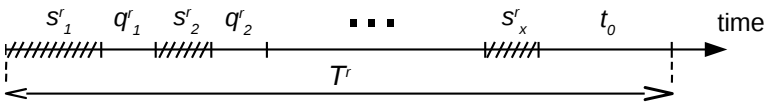


Figure 3.9: The user behaviour model describing the duration of a single data session T^r of class r.

Until now, we've modelled the system's behaviour on a call level, but now we need to look in more details on a session level, within a given call. Let us denote with λ_e the *effective normal session rate*, which we assume is a Poisson variable. This rate depends on the type

of traffic generated by the call and influences the session length. For the purpose of this paper, we will use an estimated value of λ_e for web-browsing traffic. In real networks, this value can be measured easily by the MNO.

Referring back to Fig. 3.8, we can find the service rates of a single data channel as $\mu^n = \frac{1}{\bar{T^n}}$ and $\mu^a = \frac{1}{\bar{T^a}}$. An average class-$r$ service rate of the whole cell would then be given by:

$$\gamma_r = min(\bar{n}, m) \cdot \frac{\bar{n^r}\mu^r}{\bar{n}}. \tag{3.16}$$

The average attack session duration $\bar{T^a}$ is straightforward to calculate. Since malicious calls send small amount of data, or no data at all, $\bar{T^a}$ could be obtained with the following:

$$\bar{T^a} = t_0, \quad and \quad \mu^a = \frac{1}{t_0}. \tag{3.17}$$

In the case of the average normal session duration $\bar{T^n}$, we can calculate its expected value:

$$\bar{T^n} = \sum_{i=0}^{\infty} d_i \Pi_i = \sum_{i=0}^{\infty} (\frac{i}{\lambda_e} - \bar{q} + t_0)\Pi_i \tag{3.18}$$

where d_i is the duration of T^n for i - number of consecutive burst arrivals before a timeout, Π_i is the probability of it happening and $\bar{q}$ is the average duration of the quiet period. Since inter-arrival times are exponentially distributed, the probability of i inter-arrivals I happen before a timeout could be calculated with:

$$\Pi_i = [Prob(I < t_0)]^i \cdot Prob(I > t_0) = \left(1 - e^{-\lambda_e t_0}\right)^i \cdot e^{-\lambda_e t_0}. \tag{3.19}$$

If we look at a quiet q period, it can be interrupted by either an arrival of data burst or by the inactivity timer. The intervals between burst arrivals are exponentially distributed with a mean value of $1/\lambda_e$, and the inactivity timer is an exponential random variable with mean t_0. Therefore, the quiet period q will also be exponentially distributed as the variable $min(t_0, \frac{1}{\lambda_e})$. The mean duration of the quiet period is then simply:

$$\bar{q} = \frac{1}{\lambda_e + \frac{1}{t_0}}. \tag{3.20}$$

To put it all together, we plug in the equations for $\bar{q}$ and Π_i in the equation for $\bar{T^n}$ and we get:

$$\bar{T^n} = \sum_{i=0}^{\infty} \Big[\frac{i}{\lambda_e} - \Big(\lambda_e + \frac{1}{t_0} \Big)^{-1} + t_0 \Big] \big(1 - e^{-\lambda_e t_0}\big)^i \cdot e^{-\lambda_e t_0}, \tag{3.21}$$

which further gives the normal channel service rate as $\mu^n = \frac{1}{\bar{T^n}}$.

Table 3.1: The default parameters of the model.

m=20	The available simultaneous DCH channels in a working UMTS cell depend on multiple factors. In the following work the choice of DCH channels is selected randomly because of the following reasons: i) this work is not focused on looking at the details on the physical and data link layers, and ii) the selection of parameter m only influences the scale of the results.
λ_0^n=1, λ_0^a=0.5	We keep the value of λ_0^n to a fixed random value, such that it doesn't overload the system, while the value of λ_0^a is selected to represent an attack rate that is lower than the total normal rate. In most of the experiments the value of λ_0^a is varied.
p_0^n=0.9, p_0^a=0.1	We have selected that normal users have low activity i.e., that there's 90% chance that a user will put his phone to sleep after a successful call. In contrast, there's only 10% chance for a successful attacker to leave the network, because we assume that successful attackers will keep on sending repetitive requests.
p_{b0}^n=0.9, p_{b0}^a=0.3	Normal users are selected to have low persistence i.e., lower probability of re-trying a call after being blocked. The attackers are selected with medium to high persistence indicated by the lower value of p_{b0}^a.
λ_e=0.05	The effective normal session rate would normally be estimated from usage statistics by the MNO. In our case, it is selected as the call rate to a single data channel λ_0^n/m.
t_0=2s	The inactivity timer of DCH channel is usually set to a value of around few seconds [4]. The proposed mechanism uses a low value of the timer to enable quick (de)allocation of resources.

3.3.4 Numerical results

One of the goals in the experiment in this Section was to examine the influence of the attack on the network and to identify the points under highest impact. For that purpose, we first define two performance metrics: the *signalling server load* λ_s as the total call arrival rate to the SS node, and *data channels load* λ_d as the total call arrival rate in the data channels of a cell:

$$\lambda_s = \lambda_s^n + \lambda_s^a, \quad \lambda_d = \lambda_s \cdot p_b, \tag{3.22}$$

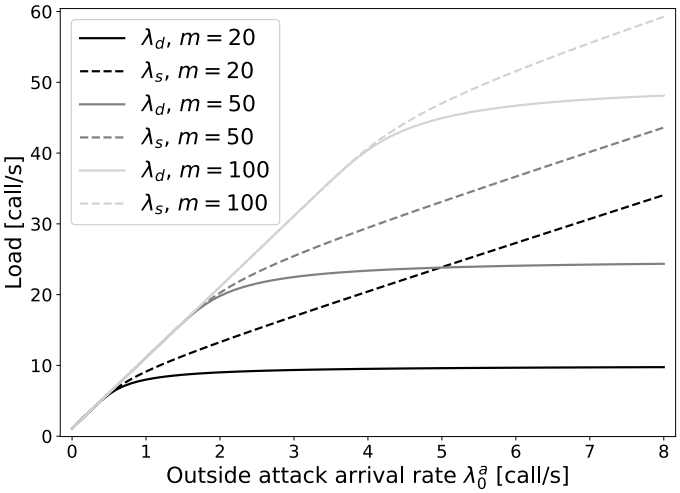


Figure 3.10: Load on the signalling server and the data channels for different attack rates.

where λ_s^r is given with Eq. 3.8. For the numerical results presented in the following, we use the default set of parameters listed in Table 3.1, unless stated otherwise for the particular experiment.

Both metrics are shown on Fig. 3.10 for different number of data channels m. We can observe that the limitation of dedicated data channels limits the load imposed on them and indirectly serves as a self-defensive mechanism. On the other side, the signalling server is the bottleneck in this situation as λ_s grows almost linearly with the outside attack arrival rate λ_{0a}. We can suppose a similar behaviour of the base station nodes, because of their similar characteristics. More detailed analysis of the bottleneck in the network, regarding BS and SS capacities is given in Chapter 4.

Another interesting observation is the influence of attack persistency on the network load. Fig. 3.11 depicts the signalling server load λ_s and the data channels load λ_d as a function of the attack arrival rate λ_0^a for the following three levels of attack persistency: low ($p_b^a 0 = 0.8$), medium ($p_b^a 0 = 0.4$), and high ($p_b^a 0 = 0.2$). While it is obvious that with the rise of attack persistency there is an increase in the signalling load in the network, it has little to no influence on the data channels load.

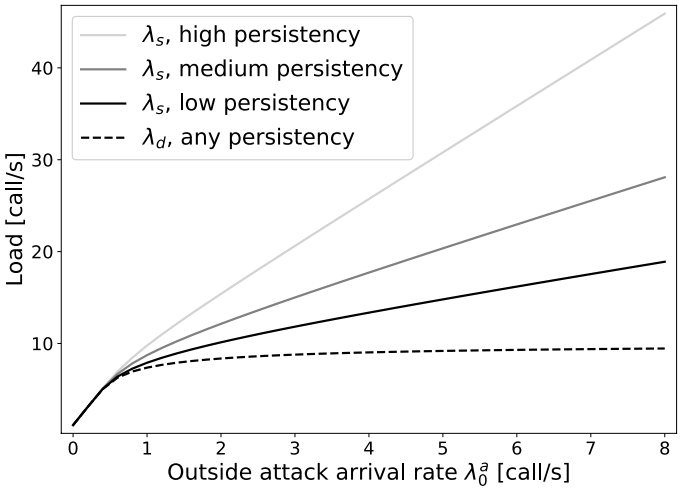


Figure 3.11: Network load for different levels of attack persistency.

Next, using the initial conclusions from Section 3.2, we hypothesise that the inactivity timer t_0, if set properly, would alleviate the problem. To see its influence on the system under attack, Fig. 3.12 shows the normalised loads λ_s and λ_d as a function of t_0. The result confirms the conclusions from Section 3.2 that a higher value t_0 could be used as a self-defensive mechanism against signalling storms. This parameter is by default set to a fixed value by the mobile operators. Anyway, with the emergence of signalling related attacks, it may be more suitable to use a dynamic inactivity timer, as a function of the network load. One possible approach is to increase the timer linearly with the load on the signalling server, after a signalling load threshold value θ is reached:

$$t_0(\lambda_s) = \begin{cases} t_0^{min} & \lambda_s \leq \theta, \\ \frac{(t_0^{max} - t_0^{min})}{\lambda_s^{max} - \theta} \cdot (\lambda_s - \theta) + t_0^{min} & \lambda_s > \theta, \end{cases}$$

where λ_s^{max} is the maximum allowed load on the signalling server, θ is a load threshold and t_0^{min} and t_0^{max} are the minimum and maximum values that the timer can take. In real operating network, these parameters should be set by the MNO according to its needs. The idea is to set the timer to a low value t_0^{min} for low signalling loads, such that bandwidth is used efficiently, and to increase the timer only after a predefined load threshold θ is reached. λ_s^{max} is the point of network outage.

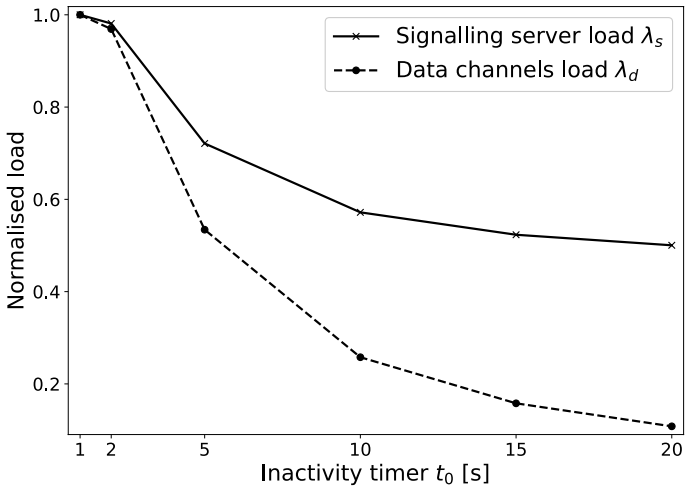


Figure 3.12: Normalised load on the signalling server and the data channels for different inactivity timers.

In the following we compare the proposed *dynamic* inactivity timer with the current *static* used in industry. The parameters used in this experiment are listed in Table 3.2, while Fig. 3.13 shows the signalling server load for both static and dynamic timers. The comparison shows that the proposed approach managed to lower the amount of signalling load for $\lambda_s > \theta$. Although this solution is not capable of evading the possibility of network outage completely, it gives MNOs control of the amount of load on the signalling server.

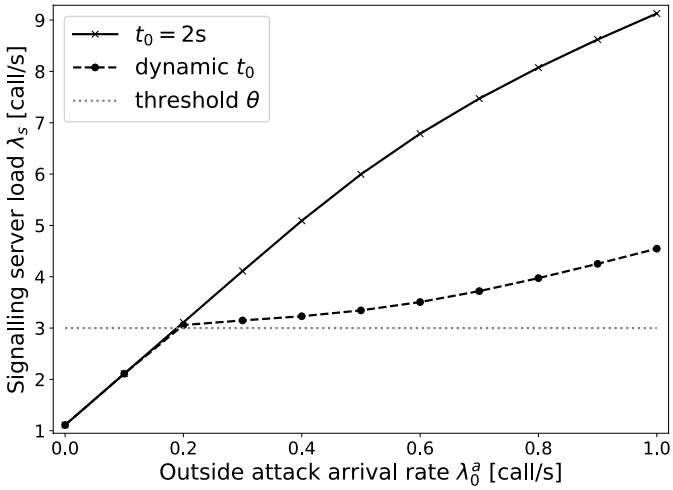


Figure 3.13: Signalling server load for static and dynamic inactivity timer.

Table 3.2: The default parameters of the dynamic inactivity timer.

Static inactivity timer	
t_0=2s	A pretty small value for the timer would enable quicker bandwidth de-allocations, and more signalling overhead.
Dynamic inactivity timer	
t_0^{min}=2s	We choose the same value as for the static timer, so for small signalling loads the network will use quick bandwidth de-allocations.
t_0^{max}=60s	The upper limit of the inactivity timer is set to a large value, a couple of times larger than t_0^{min}, so in high signalling loads the network would prefer to waste bandwidth in order to save processing power.
θ=3 calls/sec	The load threshold should represent the boundary between a "normal" and "abnormal" network loads. It is selected to a value lower than the data channels saturation load which is around 8 calls/s (see Fig. 3.10).
λ_s^{max}=5 calls/s	The maximum load the network can cope with is also selected to a value lower than the data channels saturation load which is around 8 calls/s (see Fig. 3.10).

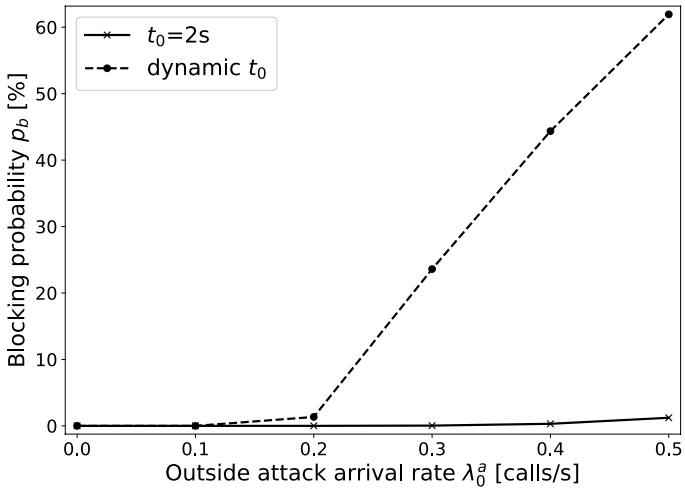


Figure 3.14: Blocking probability for static and dynamic inactivity timer.

One downside of a proposed dynamic inactivity timer is the increased portion of blocked normal calls, once the threshold θ is achieved. Fig. 3.14 depicts this situation using the blocking probability p_b as a function of the outside attack arrival rate λ_0^a. In the figure, the threshold θ is achieved around $\lambda_0^a = 0.2$ calls/s, after which the dynamic approach increases the inactivity timer. The mechanism lowers the impact of the signalling attack,

at the cost of the increased portion of blocked normal calls. This negative aspect could be reduced if the parameters of the dynamic timer are set accordingly to the MNO's call arrival statistics.

Further improvements in this approach are possible if the dynamic timer is setup for each mobile terminal individually, such that only potentially malicious terminals will be assigned long timers. As a bottom line, we can see that the inactivity timer is not just a trade-off between the bandwidth reuse and number of connections, as discussed earlier, but also in networks under a signalling attack, it is a trade-off between the signalling load in the network and the number of unserviced normal calls.

3.4 Chapter summary

The problem with overwhelming signalling messages in mobile networks is mainly created by the increasing use of smart devices and has negative effects on both the mobile network and the mobile users. This problem can happen as a consequence of poorly designed mobile applications, or because of intentionally created malware residing on mobile devices. The vulnerability of the system lies in the RRC mechanism which is designed to allow dynamic reuse of wireless communication resources, therefore allowing frequent RRC state changes. Signalling related DoS attacks are well documented in 3G UMTS networks and can cause numerous problems: overloading of core parts of a network - which is manifested through network congestion or even partial or complete network failures, high battery consumption and low computational performance for infected mobile terminals, service degradations for uninfected terminals, etc.

This Chapter, using mathematical modelling techniques from probability theory, tried to understand and tackle some of the problems. First, a mathematical model of the 3G UMTS RRC mechanism on the side of the mobile terminal was proposed. The model used Markov chains as a stochastic modelling technique which is suitable for steady-state analysis of the problem. The model consisted both normal and attacked states and a cost function was defined to minimise the probability of attacked ones. Two possible defensive mechanisms were outlined as: modifying the inactivity timers and adding setup

delay to promoting transitions. Some conclusions in this part are valuable for designing of attack mitigation mechanisms. Such mechanisms are combined with two attack detection algorithms in Chapter 5, and are implemented in a simulation environment.

In the second part of this Chapter, we proposed a queueing network model of the radio access part of 3G UMTS network. It used two classes of calls traversing an open network model and a limited number of dedicated data channels. The admission control in the network was based on the availability of data channels, which were modelled as m parallel servers without queueing possibility. The bandwidth usage patterns of normal and attack calls was also included in the model. This model confirmed some initial findings earlier in the Chapter regarding the ability of the network of self-defence using the inactivity timers and identified the bottlenecks in the network under attack.

The lessons learned in this Chapter are as follows:

- The vulnerability to signalling storms lies in the Radio Resource Control mechanism;
- Inactivity timers and call setup delays are identified as possible network parameters that could help in reducing the impact of signalling storms;
- The limitation of dedicated data channels indirectly acts as a self-defensive mechanisms in signalling-related attacks;
- The bottleneck in the network under a signalling attack are the signalling servers, and base stations, depending on their respective capacities, the number of malicious devices under control and their attack rate;
- Inactivity timers cannot prevent from network outages in signalling storms, but can decrease their impact. One possible use is the *dynamic timer* which sets the inactivity timer as a function of the load in the network;
- The inactivity timer is not just a trade-off between the bandwidth reuse and number of connections, but also a trade-off between the signalling load in the network and the number of unserviced normal calls.

Based on the network model in this Chapter, the following Chapter proposes a generic network model which is used to represent different network technologies, preceding and following 3G. It adds more details to the model and tries to quantify the impact of signalling attacks on both - the network and mobile users.

Chapter 4

Signalling storms beyond 3G mobile networks

The previous Chapter looked at the signalling problem in 3G UMTS, presenting mathematical models of the mobile terminal, and the network under attack. It examined if the system can keep its stability using some of its internal parameters, such as the inactivity timer, and identified the bottlenecks in the network. One of the shortcomings in the previous Chapter is that it looks at a single generation of mobile networks. Therefore, this Chapter improves the network model proposed in the previous Chapter, so it can represent different network generations and give more details on the impact of a signalling attack. The modelled network has a generic architecture and can provide a deeper information on both the signalling and the data communication stages, covering a typical lifetime of a call from the moment of joining to the moment of leaving the network. The rest of this Chapter is organised as follows. Section 4.1 presents the network model and describes how mobile technologies are grouped in two categories. Section 4.2 presents the user behaviour model which enables modelling of different types of user traffic. Section 4.3 describes the modelling details of the two groups of mobile technologies, while Section 4.4 analyses the network performance under a signalling storm. Finally Section 4.5 lists the conclusions and lessons learned within this Chapter.

4.1 Network model

In order to examine different network types and multiple performance metrics, the proposed network model from Fig. 3.7 is extended to a more complete and generic model, with its core part depicted on Fig. 4.1. The core part of the model consists only the basic elements of the architecture, such as multiple Base Station (BS) nodes, comprising queueing models in the control and data plane, connected to a single radio network controller, comprising one Signalling Server (SS) node. Comparing with the model in the previous Chapter, we have made two major modifications: i) we have added the control plane aspect in the BS nodes which will allow us to calculate some performance metrics related to the signalling stage, and ii) we represented the communication nodes as *black boxes*, such that different communication technologies can be plugged in as sub-models. To facilitate the analysis, different network types are assigned to one of the two mobile network groups, depending on the way they handle bandwidth allocations: *traditional* and *contemporary* groups. The traditional group consists networks which provide packet-switched services through the use of dedicated uplink and downlink radio resources, similar to the circuit-switched domain. This group includes networks such as: GPRS, EDGE and earlier versions of UMTS. The contemporary group consists networks which use a sharing approach of resources with fast scheduling, such as HSPA and LTE networks.

The lifetime of a call could be represented with two stages: a *signalling stage* where the call uses the control plane of the network as a signalling connection request message, and a *communication stage* where the call is admitted for communication and uses the data plane for data transmission. The signalling stage of the model is responsible for call admission control and scheduling according to the bandwidth resources available at the subsequent communication stage. When a mobile terminal wants to communicate, it sends a connection setup request which needs to be processed at the BS and SS. If admitted, the mobile proceeds to communicate in sessions (each comprising multiple data packets) which we denote as *calls* in the rest of the Chapter. If a call is blocked, then the mobile may either leave the network or attempt to reconnect with a probability that depends on the type of call. There are two types of calls or connection setup requests in the network: i)

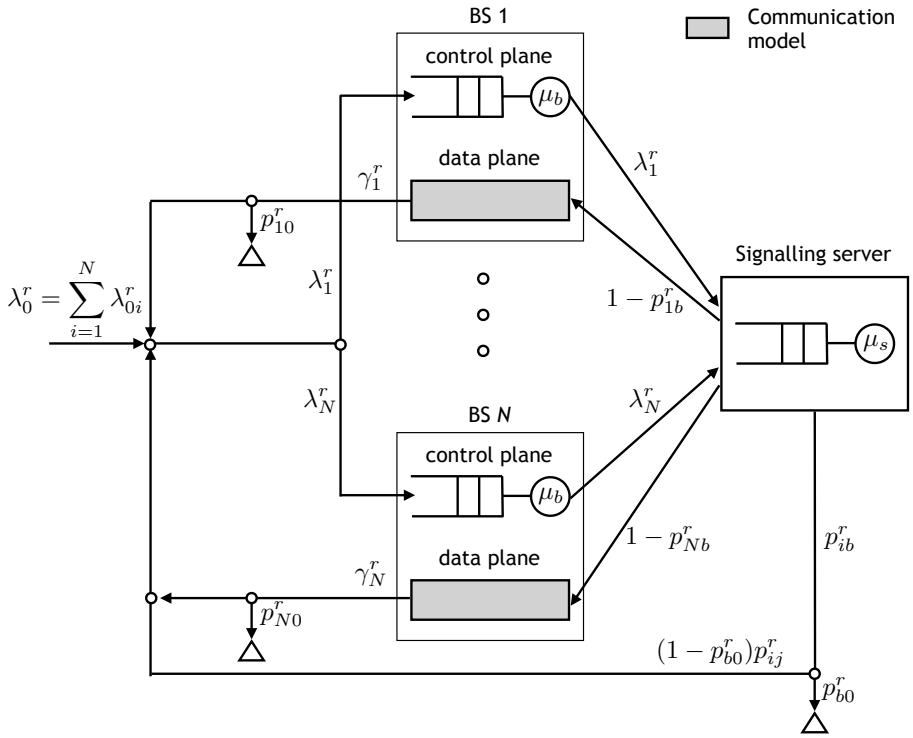


Figure 4.1: Queueing network model of a mobile network where the communication model is represented as a black box since it depends on the mobile technology.

normal calls representing traffic from legitimate users or applications, and ii) attack traffic generated by malicious or malfunctioning applications that may overload the network. The network model is open with calls joining and leaving the network, representing for example the arrival and departure of mobiles to WiFi areas. Its parameters are defined in Table 4.1 where the superscript $r \in \{n, a\}$ denotes the class of a call (normal n or attack a).

We assume calls arrive from outside the network according to independent Poisson processes and the service times in each node are independent and exponentially distributed. Since calls may be blocked at the SS due to congestion, the aggregate arrival processes at different parts of the network are not Poisson. Nevertheless, to simplify matters so as to obtain analytical solutions, we make the approximation that all flows within the network are Poisson. The service time distribution for the BS and SS nodes in the signalling stage is same for both classes of calls, because the signalling procedure undertaken by the network does not distinguish call classes. On the other hand, in the communication stage, the service time distribution is distinct for different classes of calls because of the different

Table 4.1: The main parameters of the model

N	Number of cells covered by one signalling server.
λ_{0i}^r	Rate of new class-r calls joining cell $i \in \{1, \ldots, N\}$, which corresponds to mobile phone activations and handovers by roaming users.
λ_i^r	Rate of class-r connection requests traversing the i-th BS. These include calls joining from outside the network, calls that have been successfully served and return as new calls, and calls that retry connecting after not being admitted at cell j due to insufficient data channels.
λ_s^r	Total rate of class-r calls arriving at the SS, $\lambda_s^r = \sum_{i=1}^N \lambda_i^r$.
γ_i^r	Rate of class-r calls that timed out after being admitted to cell i.
p_{ib}^r	Proportion of class-r calls not admitted for communication at cell i.
p_{b0}^r	Probability that a blocked class-r call leaves the network; p_{b0}^a represents attackers' stubbornness while p_{b0}^n reflects human persistence.
p_{i0}^r	Proportion of class-r calls leaving the network after successful service at cell i.
p_{ij}^r	Proportion of class-r calls joining cell j after being blocked at cell i given that they stay in the network i.e., $\sum_{j=1}^N p_{ij} = 1$.
μ_b	Class-independent service rate of connection requests in the BS, representing the *cell signalling capacity*.
μ_s	Class-independent service rate of connection requests in the SS, representing the *SS capacity*.
$1/\alpha^r$	Average communication time of a burst within a class-r session.
$1/\beta^r$	Average duration of a quiet (inactivity) period within a class-r session.
τ^r	Timeout rate.

bandwidth usage behaviour of the normal and malicious calls.

The flow of calls in the above model could be expressed in a closed form as follows. The total arrival rate of class-r connection requests at BS i is the sum of the rates of i) new calls, ii) returning calls that timed out, and iii) calls that were blocked at a cell j by the SS due to insufficient resources and are attempting to connect at cell i:

$$\lambda_i^r = \underbrace{\lambda_{0i}^r}_{\text{new calls}} + \underbrace{\gamma_i^r(1-p_{i0}^r)}_{\substack{\text{reconnecting after} \\ \text{timeout}}} + \underbrace{\sum_{j=1}^{N} \lambda_j^r p_{jb}^r (1-p_{b0}^r) p_{ji}^r}_{\substack{\text{joining after being blocked} \\ \text{at cell } j \text{ due to congestion}}}, \tag{4.1}$$

where the proportion of blocked calls p_{ib}^r and the rate of admitted calls that has timed out γ_i^r depend on λ_j^r, $\forall j$. These quantities are derived in the rest of this Chapter. Additionally, the following parameters are congestion-independent and their values can be estimated by MNOs through statistical observations: the external call rate λ_{0i}^r, the probability of a serviced call leaving the network after timeout p_{i0}^r, the probability of a call leaving the network after not receiving service p_{b0}^r, and the probability of a blocked call at the j-th

cell re-attempting to connect at the i-th cell p_{ij}^r. The total arrival rate of class-r calls to the SS is then $\lambda_s^r = \sum_{i=1}^{N} \lambda_i^r$.

While the signalling stage is common for different network types, the internal parameters of the BS and SS nodes, such as their capacities and service distributions, can be adjusted to the network type. Both nodes are modelled as processor sharing (PS) systems with service capacity μ_b and μ_s calls per second, respectively. If we denote with D_i the *connection setup time* (i.e., the *signalling delay*) for a user in cell i, then its average value can be calculated as:

$$E[D_i] = \frac{1}{\mu_b - (\lambda_i^n + \lambda_i^a)} + \frac{1}{\mu_s - (\lambda_s^n + \lambda_s^a)}. \tag{4.2}$$

where the both terms provide the delay in the respective nodes. In the rest of this Chapter, we concentrate on modelling the communication stage and the behaviour of normal and malicious calls.

4.2 User behaviour model

The traffic (or user behaviour) model described in the following allows us to describe a wide range of communication patterns for both normal and attacking calls. Data communication typically happens in bursts of packets, with inactivity periods between bursts that represent *thinking* or *reading* times. For example, in the case of web browsing, a user may request a web site triggering a sequence of packet downloads, then spend some time reading the web page (inactivity period) before clicking another link and starting a new download epoch, and so on. This pattern of communication also occurs in other applications like instant messaging and video streaming, with the latter being characterised by significantly longer activity periods. While Internet traffic is well-known to exhibit self-similar characteristics, here we assume that the activity and inactivity times are independent and exponentially distributed random variables whose expected values may still be congestion dependent.

The user behaviour model described in the previous Chapter, and depicted on Fig. 3.9, showed the duration of a single session T^r of class r which comprises a sequence of *service*

s_i^r and *quiet* q_i^r periods (or equivalently, activity and inactivity intervals) ending with a *timeout* interval t_0, where $i = 1, 2, \ldots, x$, and x denotes the number of bursts within the session $x = \inf\{i : q_i^r > t_0\}$. If an inactivity interval is longer than a timeout period, set by the network operator, then the connection is released, requiring the mobile device to establish a new connection in order to resume sending or receiving data. Thus, the timeout plays a crucial role in enabling MNOs to optimise the use of bandwidth, but it makes the network vulnerable to signalling storms, as summarised in Chapter 3. The attack calls causing signalling storms are characterised with repeated requests for bandwidth which are not followed by large data transmission or reception so that they are timed out quickly, triggering repeated signalling to allocate and deallocate radio channels and other resources in the network. This misbehaviour is represented by a very short activity period (indicating a small data transfer) followed by a long inactivity period to ensure that the timer expires with high probability.

Table 4.2: Effect of the user activity rate β

Value	**Normal traffic**	**Attack traffic**
$\beta \approx 0$	User with low activity; M2M communication	Signalling-intensive malware that triggers the timeout after 1 burst
$\beta \approx \tau$	User with medium activity	Malware with moderate signalling load
$\beta \gg \tau$	User with high activity	Malware that occupies bandwidth rather than causes excessive signalling

Let $\alpha^r = 1/E[s_i^r]$ denote the data service rate, $\beta^r = 1/E[q_i^r]$ the user activity rate, and $\tau = 1/E[t_0]$ the timeout rate. The parameter α^r depends on the characteristics of the network (the faster the network, the larger α^r for the same application), while β^r is assumed to be a function of the type of traffic only. However, in practice β^r can be influenced by network performance such as when users become less inclined to click on web links if the network is slow and vice versa. Such behaviour can be incorporated into the model by making the routing probabilities state-dependent, but this would unnecessarily complicate the analysis since we are interested in network conditions where the volume of normal traffic is significantly smaller than attack traffic. Table 4.2 lists examples of the values of β^r for both normal and attack calls, where in the latter the parameter can be

considered as a measure of the attackers' aggressiveness in terms of signalling traffic.

4.3 Communication models

4.3.1 Traditional networks

Technologies in the traditional cellular networks group, such as GSM and UMTS, were designed with voice services as the dominant form of mobile traffic, and consequently they assign dedicated radio resources to mobile users (e.g., the Dedicated Channel DCH in UMTS). In particular, radio channels are allocated to a given call exclusively for the entire duration of the session, denoted T^r for class r, and they are only revoked when the timeout interval expires. Although such a circuit-switched paradigm is well suited for voice calls, it is extremely inefficient for bursty data traffic.

Our model of the communication stage of a cell was presented in Section 3.3.2 with an $M/M/m/m$ queue, commonly referred to as the Erlang-B loss system [100], with two classes of calls. The bandwidth that the MNO has on disposal for a given cell is assumed to be divided into m equal portions, each representing the aggregate capacity of the dedicated channels assigned to a user on average. Queueing is not possible, which means that an incoming call is not admitted if all channels are occupied. It should be noted that decisions regarding call admission and scheduling are performed by the SS which usually has all the necessary information on data channels usage. Hence, if a call cannot be admitted into the communication stage, it is dropped immediately at the end of its service time on the SS. The model is represented with a two-dimensional Markov chain with (k^n, k^a), as in Fig. 3.8, denoting the number of normal and attack calls at the cell. To simplify the exposition, we drop the subscript i referring to the cell number in the rest of this Section so that λ^r and γ^r denote, respectively, the rate of incoming and admitted class-r calls into the cell. We denote by $\mu^r = 1/E[T^r]$ the data channel service rate for class-r calls, which is derived in the following.

In Section 3.3.2 we derived the quantities of interest, such as: the probability of i normal and j attack calls in the cell in steady-state, that is $\pi_{ij} \equiv \Pr[k^n = i, k^a = j]$ where

$0 \leq i + j \leq m$ (see Eq. 3.10), and the probability of a call being blocked due to occupied channels $p_b^n = p_b^a \equiv p_b$ (see Eq. 3.15). The blocking probabilities are identical for both traffic types because of our approximation that all flows are Poisson. The quantities p_b and λ^r are obtained by solving the system of equations (4.1) and (3.15) with the rate of admitted class-r calls to the cell being:

$$\gamma^r = \lambda^r(1 - p_b) \tag{4.3}$$

The average number of calls $E[k^r] = \gamma^r E[T^r]$ then follows from Little's theorem, or can be calculated directly from Eq. 3.13.

Average Bandwidth Occupation Time

Here we compute the average bandwidth occupation time which is equivalent to the average session duration $E[T^r] = 1/\mu^r$ defined in Fig. 3.9. The user behaviour model described in Section 3.3.3 is now modified using a simple Markov chain model on Fig. 4.2, such that can be easily adjusted for the both, the traditional and contemporary network groups. The analysis is performed by transforming the transient process of starting a session, switching between active and quiet periods and finally ending the session with a timeout into a recurrent process with three states:

- S (*service*) denotes the state where the user actively uses the allocated channel for sending and receiving data bursts within a session (e.g., downloading a web page, sending a message, streaming a video, etc.). The time spent in this state is s_ℓ, with mean $1/\alpha^r$, after which the mobile moves to state Q. As stated earlier, α depends on the network speed and can be calculated as:

$$\alpha^r = \frac{C}{md^r}, \tag{4.4}$$

  where C is the capacity of the cell in bits/s and d^r is the average burst size in bits for class-r sessions;

- Q (*quiet*) denotes the state where the allocated channel is not utilised by the user

due to end of session, or pause in the communication (reading the contents of a web site, writing a message, selecting a video for streaming, etc). From this state, the user may move to state S with rate β^r, signifying the resume of communication, or the inactivity timer may expire with rate τ and the state changes to F.

- F denotes the end of the session. However, to simplify the computation of the session's expected value, we reset the process by introducing an artificial transition from F to S with rate 1.

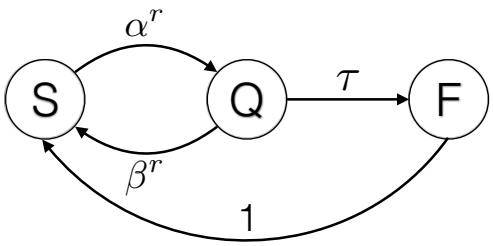


Figure 4.2: State diagram of the user behaviour model for traditional networks group.

Let us denote with Π_i the probability of the session being in one of the states $\{S, Q, F\}$, then the average session duration could be found using the following ratio:

$$\frac{\Pi_S + \Pi_Q + \Pi_F}{1 + E[T^r]} = \Pi_F.$$

Using the balance equations of the system, it is straightforward to show that the stationary solution of the recurrent Markov process is given by:

$$\Pi_S = \frac{1}{1 + \frac{\alpha^r}{\tau+\beta^r} + \frac{\alpha^r\tau}{\tau+\beta^r}},\ \Pi_Q = \frac{\alpha^r\Pi_S}{\tau+\beta^r},\ \Pi_F = \frac{\alpha^r\tau\Pi_S}{\tau+\beta^r},$$

from which we directly obtain $E[T^r]$, and the average bandwidth occupation time $E[T_B^r]$:

$$E[T_B^r] = E[T^r] \equiv \frac{1}{\mu^r} = \frac{1}{\alpha^r} + \frac{1}{\tau} + \frac{\beta^r}{\alpha^r\tau}. \tag{4.5}$$

In the above expression, one can see that when the timeout is very short, with $\tau \to \infty$, the average session duration tends to the communication time of a single burst $1/\alpha^r$. Such configuration can be energy efficient for mobile devices, since connection is released

and radio powered off if there is no traffic, but it is extremely resource intensive for the network due to excessive connection setup and release requests. Indeed, some mobile device manufacturers experimented in the past with such configuration, by introducing a non-standardised *fast dormancy* feature that disconnects a mobile device if it does not immediately have data to transmit. However, this battery-extending solution created a lot of signalling overload problems that prompted industry bodies to standardise the feature, allowing the device to resume connection with much lower overhead. Similarly, an aggressive attacker that sends a single burst and waits for the timer to expire is characterised in the user behaviour model by $\beta^r \to 0$, which after substitution in (4.5) yields an average session time of $1/\alpha^r + 1/\tau$ as one would expect.

4.3.2 Contemporary networks

More recent cellular mobile technologies, like HSPA and LTE, have abandoned the approach of assigning dedicated communication resources to active sessions, and instead implemented an approach where resources are shared among sessions using fast scheduling algorithms. In particular, bandwidth resources are allocated in small time-frequency resource blocks (RBs), which are assigned to data bursts and revoked on demand, thus reducing idle channel occupation between bursts. While standardisation bodies recommend the basic approaches for bandwidth allocation and admission control, they still vary among MNO's implementations. Furthermore, while in practice there are imperfections in the network measurements for making scheduling decisions, we will make the following simplifying assumptions to render the analysis more tractable: i) data sessions that are not sending or receiving traffic do not occupy bandwidth; and ii) the total bandwidth of a cell is shared equally among active data sessions.

Based on the above assumptions, the communication stage for a single cell can be modelled as shown in Fig. 4.3. The model consists of a PS queue that captures the physical allocation of RBs, and an abstract $M/M/\infty$ system that acts as a delay unit to represent the thinking or reading time described in Sec. 4.2. To incorporate the effect of admission and congestion control mechanisms, the PS queue is assumed to have a finite capacity

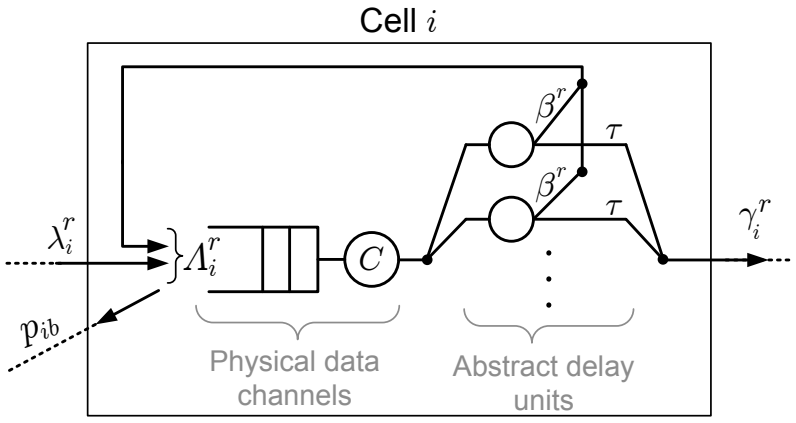


Figure 4.3: Queueing model of the communication part of a cell in contemporary networks.

m, allowing to accommodate a limited number of simultaneously active calls in order to maintain QoS. When a data burst completes service at the PS queue it joins the delay unit, from which it may again trigger a new data burst with rate β^r or end the session with rate τ. If a new call or an inactive ongoing call wants to use bandwidth (i.e., join the PS queue) but there are currently m active sessions, the call is blocked.

There could be two types of class-r calls that arrive at the PS queue: new connection setup requests, and idle ongoing calls who want to resume communication. Again dropping the reference to the cell number i, the total arrival rate of calls to the PS queue which we denote by Λ^r is given by the traffic equation:

$$\Lambda^r = \lambda^r + \Lambda^r(1-p_b^r)\frac{\beta^r}{\beta^r+\tau} = \frac{\lambda^r}{1-(1-p_b^r)\frac{\beta^r}{\beta^r+\tau}}, \tag{4.6}$$

where $\beta^r/(\beta^r+\tau)$ is the probability that an active call resumes communication (sends or receives a burst) before its timer elapses, and so the second term in the sum represents the fraction of admitted calls which attempt to resume communication after at least one quiet period.

Admitted users are no longer guaranteed a certain data rate like the Erlang-B model of traditional networks, and it should be emphasised that the average duration of the active periods $1/\alpha^r = E[s_i^r]$ in Fig. 3.9 is now a function of bandwidth congestion at the cell. Instead, calls experience lower QoS as cell load increases (up to a certain level because of the maximum occupancy constraint m). Consequently, $1/\alpha^r$ represents the average

response time of class-r calls in the PS queue, while $1/\mu^r$ denotes its *service demand* which depends on the cell capacity C in bits/s and the size d^r of the data transferred per burst:

$$\mu^r = \frac{C}{d^r}. \tag{4.7}$$

Note that μ_r takes a form similar to α^r in the traditional network model, which was derived in Eq. 4.4.

The state of the PS node is described by the pair (k^n, k^a) denoting the number of normal and attack calls that are active, and its stationary probability $\pi_{ij} \equiv \Pr[k^n = i, k^a = j]$ is given by [101, 102]:

$$\pi_{ij} = \binom{i+j}{i} \rho_n^i \rho_a^j \pi_{00}, \quad 0 < i + j \leq m,$$

where $\rho_r = \frac{\Lambda^r}{\mu^r}$ and:

$$\pi_{00} = \Big[\sum_{i,j=0}^{m} \binom{i+j}{i} \rho_n^i \rho_a^j \Big]^{-1}.$$

The probability of a new or ongoing call being blocked is:

$$p_b = \sum_{i=0}^{m} \pi_{i(m-i)} = \sum_{i=0}^{m} \binom{m}{i} \rho_n^i \rho_a^{m-i} \pi_{00}. \tag{4.8}$$

Furthermore, the rate of calls that have timed out after completing at least one service at the PS queue is:

$$\gamma^r = \Lambda^r (1 - p_b) \frac{\tau}{\beta^r + \tau}. \tag{4.9}$$

The values of p_b, Λ^r, and λ^r are subsequently obtained numerically by using the above expression for γ^r and solving the system of equations 4.1, 4.6 and 4.8.

The average number of calls occupying the data channels can be calculated as:

$$E[k^n] = \sum_{i=1}^{m} i \sum_{j=0}^{m-i} \pi_{ij}, \quad E[k^a] = \sum_{j=1}^{m} j \sum_{i=0}^{m-j} \pi_{ij}, \tag{4.10}$$

from which we directly obtain the average time it takes a class-r traffic burst to complete communication as:

$$\frac{1}{\alpha^r} = \frac{E[k^r]}{\Lambda^r (1 - p_b)}. \tag{4.11}$$

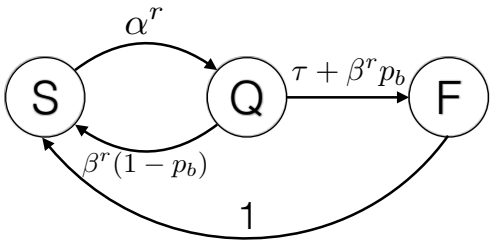


Figure 4.4: State diagram of the user behaviour model for contemporary networks group.

Finally, the average session duration can be estimated *approximately* using the recurrent Markov approach developed in Section 4.3.1, with a modified transition rates matrix (see Fig. 4.4): α^r and the superfluous rate 1 for, respectively, the transitions $S \to Q$ and $F \to S$ as before; $\tau + \beta^r p_b$ for $Q \to F$ so that a call may exit due to either timeout with rate τ or being blocked with probability p_b after attempting to transfer a burst with rate β^r; and $\beta^r(1-p_b)$ for $Q \to S$ to reflect the fact that only a fraction $(1-p_b)$ of bursts from ongoing calls is allowed to communicate. This leads to:

$$E[T^r] = \frac{1 + \frac{\tau+\beta^r}{\alpha^r}}{\tau + \beta^r p_b}. \tag{4.12}$$

The bandwidth holding time $E[T_B^r]$, however, is no longer equal to $E[T^r]$ as in the traditional networks, since bandwidth of contemporary systems is mostly consumed while sessions are active. Thus, the average time a session occupies bandwidth is the fraction of the session duration spent in state S:

$$E[T_B^r] = \Pi_S E[T^r] = \frac{\tau + \beta^r}{\alpha^r(\tau + \beta p_b)\left[1 + \frac{\alpha^r(\tau+\beta p_b)}{\alpha^r+\tau+\beta^r}\right]}, \tag{4.13}$$

where Π_S is calculated from the modified state diagram in the same way as for the traditional networks.

4.4 Performance evaluation

In this Section, we compare the behaviour of the two groups of networks as the arrival rate of misbehaving calls λ_0^a is varied, under the same conditions (across different technologies) regarding cell bandwidth B, number of channels per cell m and arrival rate of normal calls

λ_0^n and assuming homogeneous cells.

The proposed model can be used to evaluate several performance measures of interest, including the signalling delay $E[D_i]$ given in (4.2); blocking probability p_b of (3.15) and (4.8); communication delay for normal calls, expressed as $1/\alpha^n$ in (4.4) and (4.11); and bandwidth occupation time of a session in (4.5) and (4.13) which we denoted by $E[T_B^r]$.

4.4.1 Mapping the models to 3GPP systems

We discuss in this Section how the parameters of the models are obtained. The data capacity C of a cell (bits/s) can be estimated from the *spectral efficiency* η which is defined as the number of bits/s/Hz supported by the corresponding technology, and by the bandwidth B (Hz) available to the service provider, that is:

$$C = \eta \times B. \tag{4.14}$$

Table 4.3 lists, for different 3GPP generations, typical values of η measured from operational networks [103] and the corresponding capacities when $B = 2 \times 20$ MHz. Note that the values of C shown in the table for 2G and 3G cells do not reflect their actual capacities, since these networks operate on smaller bandwidth (e.g., 10MHz with UMTS).

Table 4.3: Estimated cell data capacity

Technology	η [**bit/s/Hz**]	**C** [**Mbps**]
2G+ GPRS (Rel 97)	0.04	1.6*
3G UMTS (Rel 99)	0.19	7.6*
3G+ HSPA (Rel 6)	0.68	27.2
4G LTE (Rel 8)	1.32	52.8
4G+ LTE-A (Rel 10)	2.6	104

* Scaled up according to $B = 2 \times 20$.

The signalling capacities of the BS and SS depend on the configurations set by the MNOs. In any case, there are major differences between traditional and contemporary networks concerning where RRC signalling is performed. Indeed, BS in GSM/UMTS are responsible for a small part of the connection setup and release procedures, acting primarily as dumb

relays, while a substantially bigger part is managed by the radio network controller (RNC) which represents the SS in our model. On the other hand, LTE distributes most of the RRC overhead among BS and does not have RNC in the access network, and uses a dedicated SS in the core network called the mobility management entity (MME). Thus, while the MME handles much fewer RRC signalling messages than the RNC, offsetting some of the load to more intelligent LTE BS, it is still vulnerable to signalling storms due to the massive number of BS and mobiles under its control.

An estimate of the cell signalling capacity μ_b for each technology can be found from the corresponding capacity of the Random Access Channel (RACH). For an LTE normal-sized cell, with high traffic demand, a RACH preamble sequence could be transmitted three times per 10 ms frame, giving a signalling capacity of μ_b=300 calls per second [104]. The same approach is valid for newer generations of LTE. With the traditional technologies, however, the calculation is less straightforward, so some vendors simply advise to calculate or dimension the cell signalling capacity as a fraction (<10%) of the data capacity [105]. Thus, in our evaluation of UMTS, we calculate μ_b based on 5% signalling capacity and a 2KB average cumulative size of signalling traffic, yielding $\mu_b = 0.05C/2KB = 23.75$ calls per second. Similarly, the capacity of the corresponding SS are scaled with the number of cells i.e., $\mu_s = N\mu_b$. Table 4.4 shows the values of the other parameters used in our numerical results, some of which are based on industry reports as described above. Note that in real networks, however, the network parameters are dimensioned by MNOs according to their needs.

Table 4.4: Values used in the numerical results

Parameter Value	**Remarks**
τ=0.2	Common timeout value in real networks
m=20	Approximately 384 kbps per channel in UMTS
$\lambda_0^n = 1$	Fixed external arrival rate of normal calls
p_{b0}^n=0.9, p_{b0}^a=0.3	Attackers are more stubborn than normal users
p_{i0}^n=0.9, p_{i0}^a=0.1	Attacks are more likely to reconnect after timeout
d^n = 320 KB, d^a = 20 KB	Average size of a web page and of an attack burst
β^n = 0.05, β^a=0.15	Average reading time for web browsing; medium-to-high aggressive attack
$N = 100$	Percentage of attacked cells = 70%

4.4.2 Numerical results

This Section first compares two candidates of the traditional and contemporary network groups: UMTS Rel 99 and LTE Rel 8, then uses the model to examine the effect of the signalling storms on different types of traffic, and finally the UMTS mathematical model is compared with a simulation run in SECSIM.

Comparison of traditional and contemporary networks

In order to illustrate the differences in performance between the two groups, we focus on a representative generation from each group, namely UMTS (Rel 99) from the traditional networks, and LTE (Rel 8) from the contemporary networks. We then present results comparing the different technologies listed in Table 4.3 with respect to a single metric that illustrates the main design differences between the two groups.

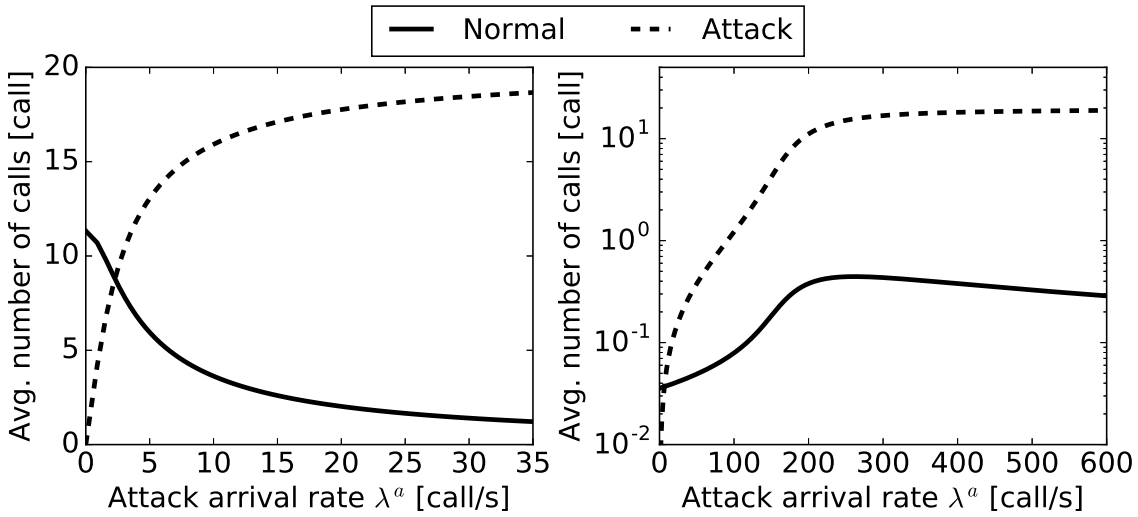


Figure 4.5: Average number of normal and attack calls concurrently occupying bandwidth in a UMTS (left) and LTE (right) cell, where smaller numbers reflect superior performance.

Fig. 4.5 shows the average number of normal and attack calls occupying the bandwidth of a cell in UMTS (left) and LTE (right), versus the attack rate λ^a. Note first that calls, regardless of their type, compete for two finite bandwidth resources: the maximum number of concurrent calls m that can be supported by a BS and the cell's data capacity C. Thus, for UMTS we see that as the attack rate increases more normal calls get pushed out of the cell. However, the situation is different and more interesting with LTE where initially the

average number of normal calls grows with λ^a, reflecting the increased contention for C and in turn the longer times that calls take to complete, but as λ^a exceeds a certain value we observe that attackers start to replace normal users in the PS system (i.e., m becomes the bottleneck), causing the average number of normal calls to drop.

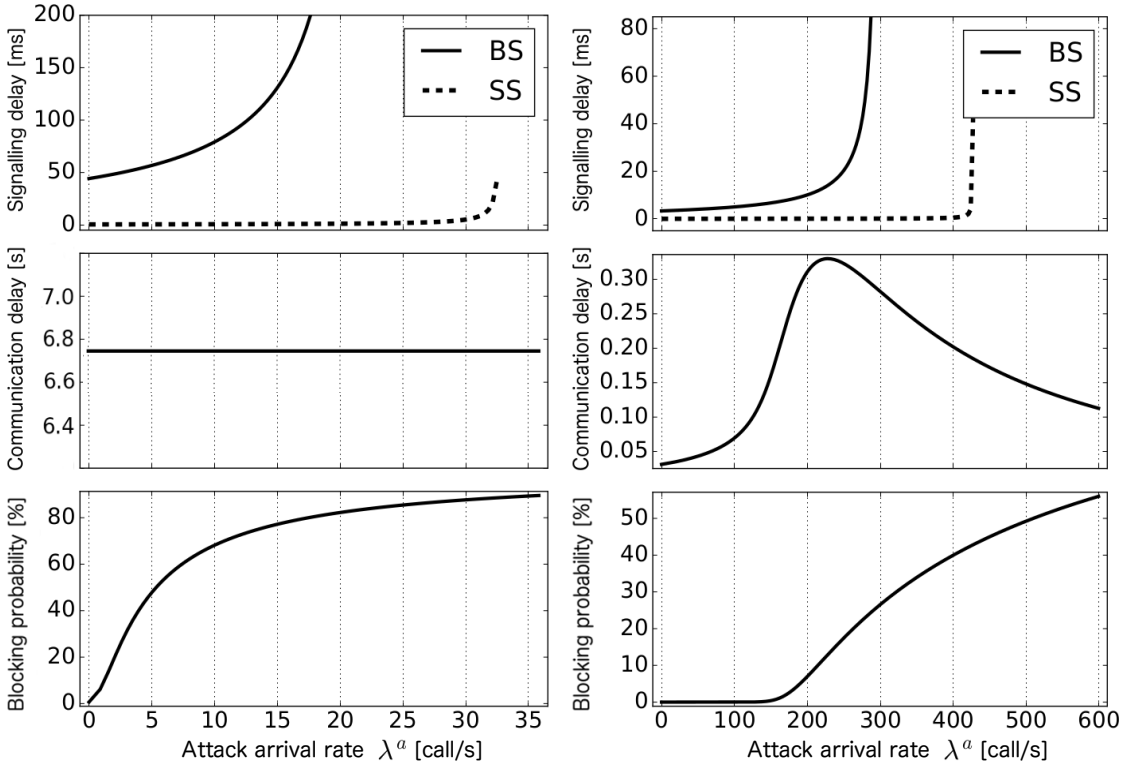


Figure 4.6: Signalling delay, communication time and blocking probability versus the arrival rate of attack traffic for UMTS-Rel'99 (left) and LTE (right).

Fig. 4.6 shows how the different QoS metrics are affected in UMTS and LTE when the arrival rate of attack traffic is varied. In the case of UMTS, as λ^a increases the signalling delay also increases, first gradually and then exploding once the signalling capacity of either the BS or the SS is reached. The results suggest that the failure of the SS requires a much higher *aggregate* attack rate, but in general the attack does not necessarily saturate the BS before the SS, in the same way distributed denial of service attacks (DDoS) affect downstream and upstream servers on the Internet. However, differently from data plane attacks, we see that the communication time of normal users in UMTS remains constant, regardless of the attack rate, because of the channel reservation scheme used in those systems. In particular, during a signalling storm, mobile users find it extremely difficult to obtain a connection but once successful they experience reasonable QoS – a phenomenon

which has been observed [106] on real network measurements taken during crowded events.

On the other hand, in the case of LTE, the network can sustain a much higher attack rate before signalling congestion or blocking probability becomes unacceptably high. Another major difference between LTE and UMTS is demonstrated with the average communication time: when the attack rate increases, normal users experience higher communication delays due to sharing bandwidth with more misbehaving mobiles. However, as the attack rate grows beyond a certain value we see that communication time drops steadily. This is due to normal users being pushed out of the system by attackers who have lower bandwidth demands (as shown in Fig. 4.5), and so the few normal users that manage to access the bandwidth will share it with less active mobiles; in turn, normal users experience shorter communication times as λ^a increases but they may get blocked shortly by the congestion control mechanism. Therefore, the negative effects that signalling storms have on normal users are long connection setup times and high blocking probability.

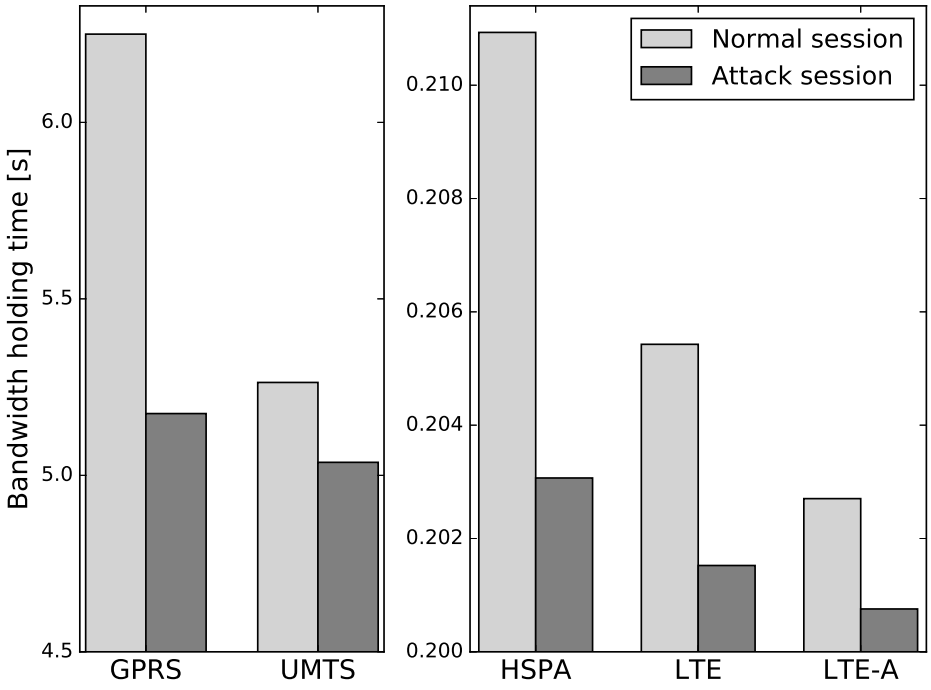


Figure 4.7: Average bandwidth holding time for normal and attack sessions in the traditional (left) and contemporary (right) networks, for attack call rate $\lambda_0^a = 1$.

To summarise, there are major differences in performance and stability between traditional and contemporary mobile networks. Among all the factors responsible for these

differences, our analysis highlights two: capacity and fast scheduling. Recent mobile networks provide higher capacities for the same amount of bandwidth per cell, because of their higher spectral efficiency which in turn depends on modulation, channel coding, antenna configuration, etc. The second factor is even more important and is underlined in Fig. 4.7 which plots the average bandwidth holding time per session for different 3GPP networks. In traditional networks (left) users tend to hold the allocated bandwidth for a prolonged period of time that includes multiple data transfer and quiet periods followed by a relatively long inactivity timeout interval. In contrast, contemporary networks employ intelligent scheduling algorithms in order to quickly recycle unused radio channels so as to reduce bandwidth wastage. This directly improves the network's resilience to signalling storms.

Comparison among different traffic types

In this part we use the proposed model to examine the effect of the signalling attacks on the traffic generated by different types of normal applications: web browsing, video streaming and machine-to-machine. The parameters of the model are set as in Table 4.4, except for the session data size d^n and the user activity rate β^n which are listed in Table 4.5. Using the same approach as earlier, we show results of the comparison among the modelled UMTS Rel 99 network of the traditional group, and LTE Rel 8 network of the contemporary group. The three normal traffic types are examined one-by-one and independent of each other, and have a fixed rate of arrival $\lambda^n = 1$, while there is an ongoing signalling attack in the network with medium-high aggressiveness $\beta^a = 0.15$.

Table 4.5: Values used in the comparison of different traffic types.

	β^n	d^n
Video streaming	2	10 MB
Web browsing	0.05	800 KB
M2M	0.001	20 KB

Two performance metrics of interest, communication delay and blocking probability, are depicted on Fig. 4.8. One general conclusion is that normal traffic with smaller session data size and smaller user activity rate are less affected by signalling storms, shown by the

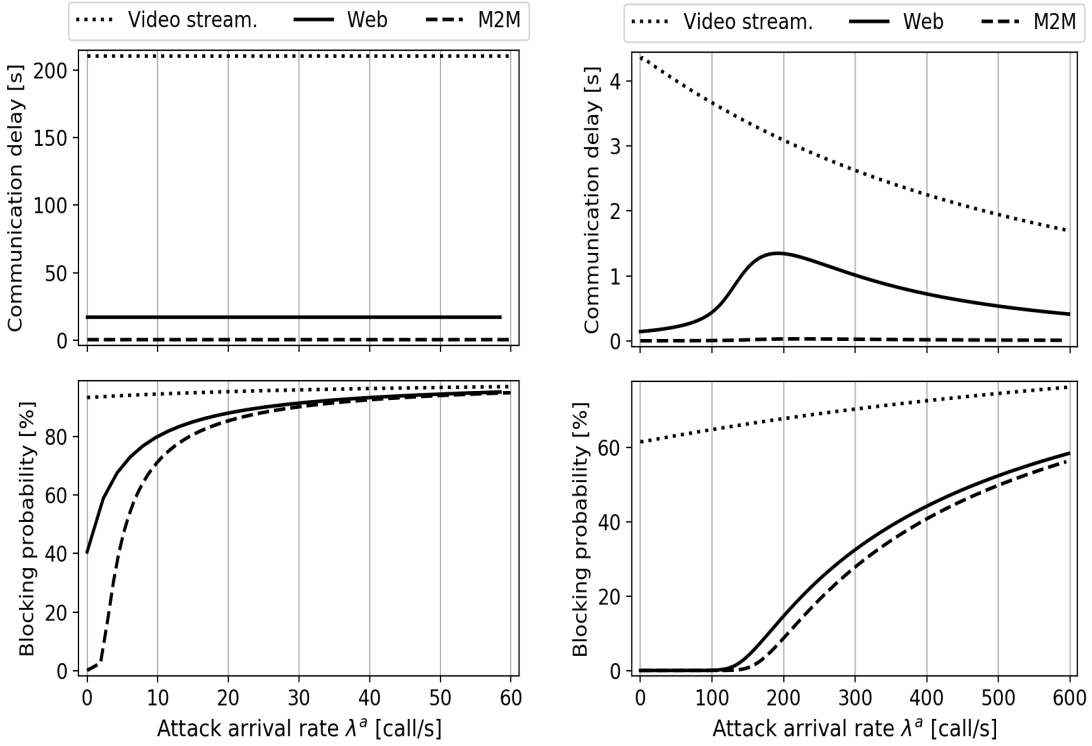


Figure 4.8: Communication delay and blocking probability for UMTS Rel 99 (left) and LTE Rel 8 (right) using different types of traffic.

lowest values obtained for both blocking probability and communication delay, for both UMTS and LTE. As expected, the communication delay in UMTS is constant for each traffic type due to the use of dedicated channels, and traffic types with larger data volumes per session experience longer delay. The blocking probability increases with the increase of the attack arrival rate for all traffic types, for both UMTS and LTE. It is interesting to note that in a UMTS cell, a new video streaming of 10 MB data in an interval of $1/\lambda^n = 1$ s, produces a blocking probability of nearly 100% without the presence of signalling storm (for $\lambda^a = 0$) i.e., the cell cannot cope with such volume and rate of data. In the case of LTE, the blocking probability for video streaming with such volume and rate, and $\lambda^a = 0$ is around 60%, and increases slowly with the increase of the attack rate. Finally, communication delay should not be a big concern for LTE, especially for M2M traffic, as delays reach up only to a few seconds in a signalling storm, while the blocking probability values get to more substantial levels. The communication delay for video streaming traffic even drops with the increase of the attack rate. This is probably due to the fact that unblocked video streams compete for bandwidth with small attack

packets (with the increase of the attack rate), rather than competing with other large volume video streams (for lower values of the attack rate).

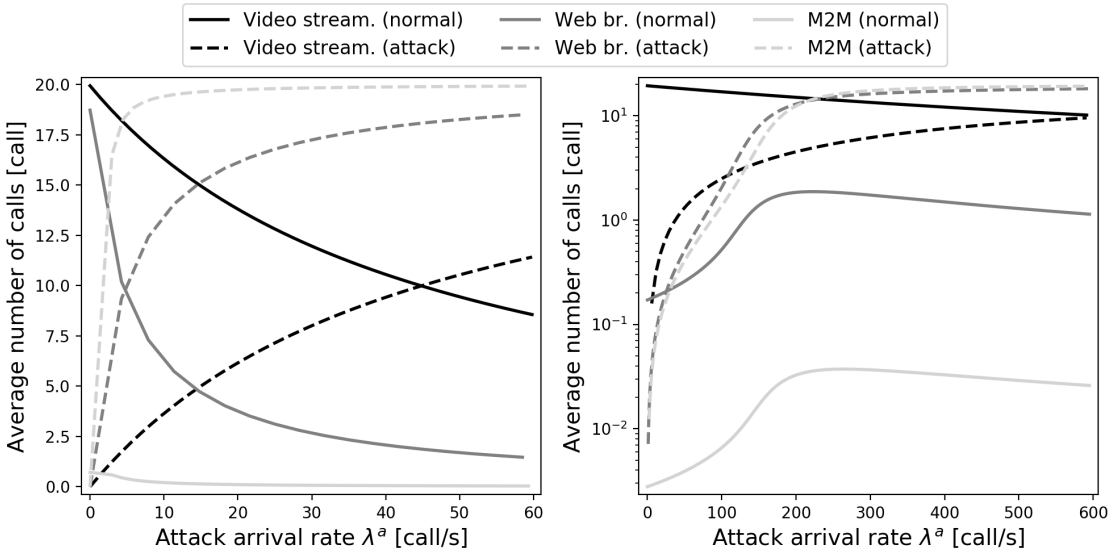


Figure 4.9: Average number of normal and attack calls in service in a UMTS (left) and LTE (right) cell, using different types of traffic.

Although according to the communication delay and blocking probability figures, networks with M2M traffic seem to be "more resistant" to signalling storms, the results on Fig. 4.9 suggest a different conclusion. Fig. 4.9 shows the average number of normal and attack calls in service in a UMTS and LTE cell for a range of values for the attack rate. If we look at the attack rate for which the number of attack calls reaches over 90% of the available resources (m=20) in UMTS, we can conclude that the the attack is easiest to spread in a system with M2M normal traffic, then web browsing, then video streaming. The important conclusion here is that although the M2M traffic itself seems to be the least impacted by the storm (according to Fig. 4.8), the mobile network with M2M traffic is more prone to being congested with signalling storm calls. The results for LTE suggest similar conclusions.

Comparison with UMTS simulation

In this Chapter we use the SECSIM simulator, described in Section 2.3.2 to compare the results gathered from our model with the simulation results. At the moment of writing, the simulator hasn't fully implemented a 4G LTE network, therefore the comparison could

only be done in terms of the UMTS system. The first step is to adjust the fixed parameters of the model and the simulation. Within this part, we made the following adjustments:

- The simulator does not support an open-network model with variable number of mobile terminals which randomly join and leave from/to areas out of the simulated network. To address this issue, we managed to setup the simulator with a fixed number of terminals, with their points of activation adjusted such they resemble a Poisson flow of call arrivals from outside the network (identical to the one used in the model), and their deactivation resembles the calls departure times. In this context we modified the probabilities of calls leaving the network to a value calculated for the simulation: $p^a_{i0} = p^n_{i0} = 0.32$.

- One of the biggest differences between the model and the simulator is in the data capacities of network nodes i.e., the simulator uses unlimited bandwidth in the radio access part of base stations, while the model uses finite capacities, as explained in Table 4.3. Part of this issue is addressed with setting $p^n_{b0} = b^a_{b0} = 1$ in the mathematical model such that blocked calls will not create an extra rate of recurring calls within the network. Other aspects of this issue require bigger modifications within SECSIM, and are left for future work, described in Section 6.2.

- A typical session in the web browsing traffic model in the simulator would contain downloading of random number of web page elements (text content, pictures, html, style sheets, etc.), with a total average web page size of 817 KB. This adjustment was inserted in the mathematical model by setting $d^n = 817$ KB, which further triggers $\alpha^n = 0.058$, calculated with Eq. 4.4. Furthermore, the inactivity parameters of the normal web browsing and the attack traffic were estimated from the simulator and set in the model as $\beta^n = 0.705$ and $\beta^a = 5.058$.

Having adjusted the web browsing and the fixed network parameters, we can visualise some of the performance metrics of interest. Fig. 4.10 shows the delay that signalling messages experience within the signalling server. The figure suggests that both curves of the simulation and the model have a similar form, and that the simulation values are higher than the model values for an order of few seconds. This is because the service rate

of the signalling server is different for the simulation and the model i.e., in the model we used an estimated service rate (which was calculated using a BS signalling capacity of 5% of the total BS capacity, and a signalling message size of 2 KB), which does not fully resemble the service rate in the simulation.

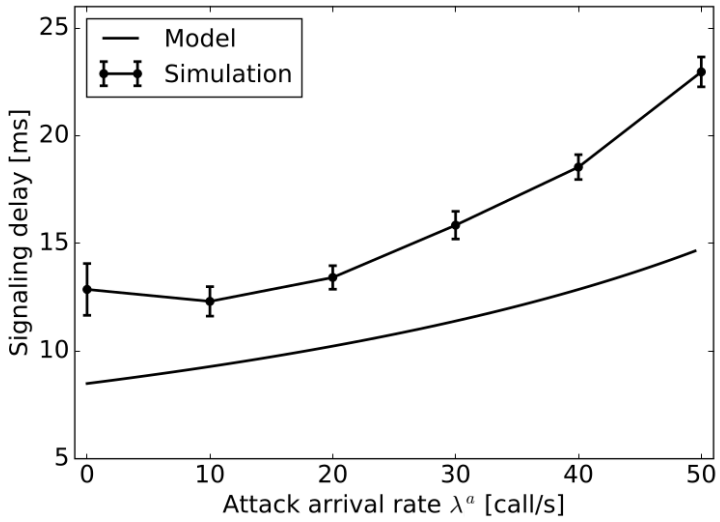


Figure 4.10: Delay in the signalling server (99% confidence interval used).

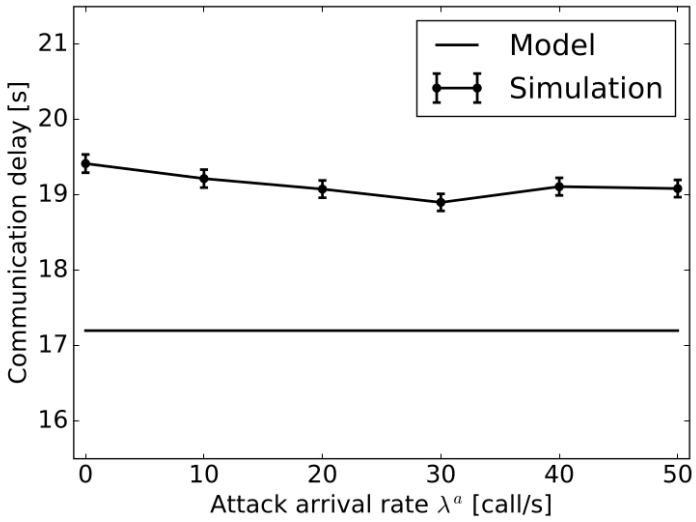


Figure 4.11: Communication delay for downloading a web page with an average size of 817 KB (99% confidence interval used).

In Fig. 4.11 we depict the communication delay for downloading a web page with an average size of 817 KB i.e., the response time measured on application layer, from the moment of generating the page request, until receiving the full web page content. Again

the simulation and model results are similar, with the simulation values being higher for an order of few seconds. This should be because of the difference in the capacities of the physical wireless channels: the simulation uses an infinite capacity for the radio access part and assigns channel data rates from a probability distribution, while the model uses a fixed wireless capacity obtained from literature, as in Table 4.3.

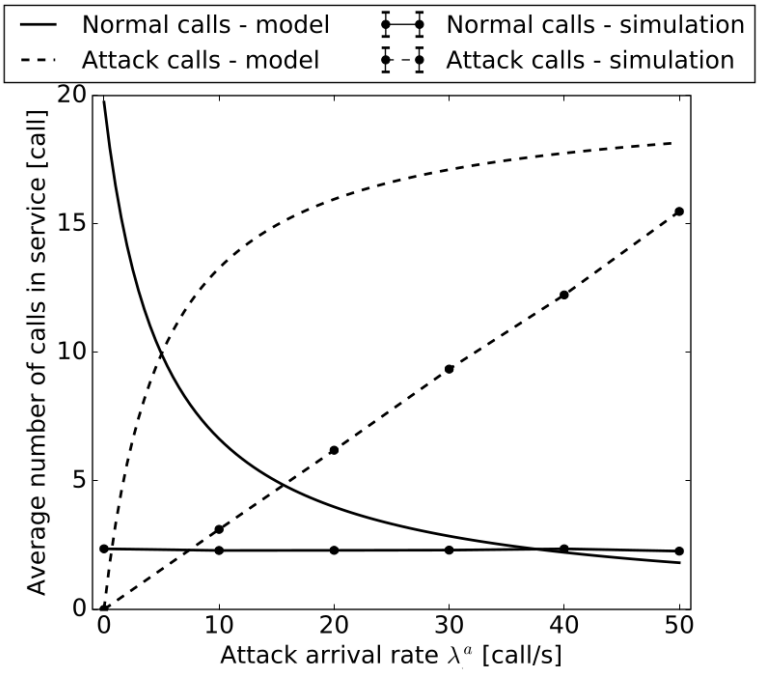


Figure 4.12: Average number of normal and attack calls in service in a single BS (99% confidence interval used).

Lastly, we were interested in the average number of concurrent calls in service in a single base station, depicted on Fig. 4.12. In this case there is a bigger discrepancy between the simulation and mathematical results. The average number of attack calls in service increases with the rise of the attack rate λ^a for both the mathematical model and the simulation. Anyway, there is a finite limit $m = 20$ for the attack calls in the model, while the attack calls in the simulation rise linearly to infinity. Furthermore, the simulation shows almost constant number of normal calls in service, independent of the attack rate, while in the mathematical model the normal calls are pushed out by attack ones and their number decreases with the increase of the attack rate. We consider that both discrepancies are due to the unlimited wireless resources in the simulator, and because of the needed time for implementing these modifications we will work on this issue in the future, as

discussed in Section 6.2.

4.5 Chapter summary

Mobile networks are becoming the access networks for much of the communication system infrastructure of the world, and this massive penetration of mobile devices poses many new problems related to security. One threat, known as a signalling storm, overloads both the network's signalling capacity at the cell level, as well as the network servers that are used for establishing, maintaining and releasing connections.

This Chapter developed a mathematical model for a mobile network with both normal and misbehaving mobile terminals that repeatedly request bandwidth but quickly time out because of inactivity, causing a radio resource control signalling storm. The model was utilised to compare the behaviour and performance of two groups of 3GPP mobile technologies: *traditional* networks that incorporate mobile data services using concepts from the circuit-switching domain, and *contemporary* networks which allow bandwidth sharing among users with fast switching. The analysis was conducted i) by abstracting many of the lower protocol stack details; ii) by making simplifying assumptions such as identical channel conditions for all users, equal sharing of bandwidth, homogeneous traffic characteristics within the same traffic class, and perfect scheduling in contemporary networks; and iii) by assuming all flows inside the network to be Poisson.

The lessons learned in this Chapter are as follows:

- Mathematical modelling techniques are useful in analysing individual parts of complex computing systems, such as mobile networks. Anyway, with the increase of complexity of the modelled system, certain approximations must be made in order to facilitate the analysis. Furthermore, such models would not always result in a closed form solution, which therefore would require the use of numerical analysis.
- The shift in the design of the radio access system, from using dedicated channels to shared ones, has made mobile networks more resistant to signalling storms. Nevertheless, 4G/LTE networks are still vulnerable to signalling storms.

- A failure in the control part of the mobile network could happen in different locations / nodes (base stations, signalling servers, etc.), and depends on: the signalling capacities of each node, the number of mobile terminals under control of the node, the number of malicious terminals in control of the node, the rate of attacking signalling requests, the design of the radio access system in the data plane and others.

- In mobile networks of the traditional group, a signalling storm would result with high blocking probability for legitimate mobile users, while in networks of the contemporary group, it would first result in service degradations (increased communication delays, dropped packets,...), and afterwards in increased blocking probability.

- The comparison of three normal traffic types (video streaming, web browsing and machine-to-machine) in a network under signalling storm suggested that attacks most easily reach the bandwidth capacity in networks which usually use small data volumes and small rates, such as in M2M. On the other side, normal users with this traffic type would experience lower service degradations (communication delay and blocking probability) in a network under attack.

- Increasing the signalling capacity is not an effective long-term solution, because signalling traffic is outpacing the exponentially growing mobile data traffic. These trends are expected to continue, especially as M2M and IoT applications transition to mobile broadband.

In outlook for the future, standardisation bodies are trying to solve the signalling overhead problem generated from M2M and other small data transfer devices through efforts such as [107, 108, 109, 110, 111]. Some of those proposals have already been included in Release 10-12 specifications. However, most of the solutions require devices to be pre-registered with the network or to declare their intent to transmit small packets; they may not prevent signalling storms from *chatty* mobile applications that do not utilise the proposed features to reduce their signalling load. Therefore, the signalling overload problem is expected to remain an open question, while active and real-time detection and mitigation systems are needed to protect future generation networks. The following Chapter proposes two such mechanism, based on the lessons learned so far.

Chapter 5

Detection of signalling storms

Besides the mathematical modelling, simulation is another approach towards addressing security related research in mobile networks. This Section uses the Mobile Networks Security Simulator (SECSIM), described in Section 2.3.2, which is a mobile networks simulator specialised in modelling, simulating and evaluating cyber-security aspects of mobile networks. Within this Chapter, SECSIM is used to implement and evaluate two signalling attack detection mechanisms. The first mechanism is based on counting repetitive connections and applying a pre-defined threshold on the count, while the second mechanism monitors bandwidth used by mobile terminals, calculates a cost function, and makes a decision by comparing the cost function to pre-defined thresholds. The ideas come from some previous research in [63] where a theoretical approach of the counter detection is presented, and [61] where our analytical and simulation work confirms that terminals performing signalling attacks don't use efficiently the allocated bandwidth, in order to evade traditional flooding detection mechanisms. Both mechanisms are implemented and evaluated using a 3G UMTS model in SECSIM. Finally the detection mechanisms are used in parallel with a simple attack mitigation approach and results show its impact on the mobile network stability and quality of service provided to customers. The counter-based detector is described in Section 5.1, while the bandwidth usage-based one in Section 5.2. Section 5.3 concludes the work in this Chapter and looks at possible future improvements in the area.

5.1 Counter-based detector

The proposed mechanism enables detection and mitigation of signalling attacks and storms per mobile terminal in real-time. It is based on counting the repetitive bandwidth allocations of same channel type (ex. a shared FACH or dedicated DCH channel in UMTS network). From an implementation perspective, it is important that the mechanism could be implemented on both mobile terminal and network/operator side. If implemented on the mobile terminal side, due to the terminal's limited resources, some special requirements are needed so it does not impose any processing, storage, and memory difficulties to the terminal. For this purpose, the proposed mechanism is envisioned as lightweight background process requiring only two input parameters: the time instances of bandwidth allocation and the type of bandwidth allocation, which are stored in memory for the duration of a time window of length t_w.

A decision of an attack being detected is simply taken when the number of repetitions reaches a predefined threshold called *counter threshold* - n. Repetitions are counted in a sliding time window manner, where the length of the window t_w is chosen according the threshold n. If we denote with t_I the duration of the inactivity timer in the attacked state, then obviously t_w is selected such that $t_w > n \cdot t_I$ i.e., the window should be large enough to collect n repetitions. The upper limit of t_w is set according the memory and storage capacities of the device on which it is implemented. While this Chapter looks at the effect of different thresholds values, research in [63] analyses the problem from an analytical perspective and shows a way of finding the optimal threshold n. Within the simulation environment we could combine the detector with an attack mitigator. The idea for attack mitigation is based on the conclusions in Sec. 3 where the attack impact was lowered by adding delay to signalling messages asking for bandwidth allocation. In a similar fashion, we will use a fixed time duration t_b called *blocking time* in which all the communication of a detected misbehaving terminal will be blocked.

In this Section we will use the SECSIM simulator to: (i) evaluate the detector in Sec. 5.1.2, and (ii) simulate an attack scenario including detection and mitigation of attacks in Sec. 5.1.3. The simulation scenario is same for both cases except that the mitigation

mechanism is switched on in the latter case.

5.1.1 Simulation setup

Our simulation setup looks at a part of a mobile network covered by a few UMTS base stations, under a single radio network controller. All mobile terminals are functioning regularly using web browsing application, while an attacker manages to compromise a portion of them and install a second (malicious) application which triggers malicious communication. The web browsing application is modelled using stochastic techniques, according to statistical distributions of real world Internet traffic [93]. The malicious application is present on 20% of terminals, a portion which has showed as big enough to overload signalling related nodes in the network in [61], and attacks on the dedicated DCH channels in the network. The attacker is assumed to have estimated the length of the inactivity timer in DCH channels with an exponentially distributed error with mean value of 3 seconds. The counter detector uses different values for the counter threshold n, and the detector window is selected as $t_w = 3nt_{DCH}$, a value which has showed as suitable in the simulations. Each simulation is run multiple times with a different seed for the random number generators, so results can show the averaged values of the performance metrics or interest.

5.1.2 Detector evaluation

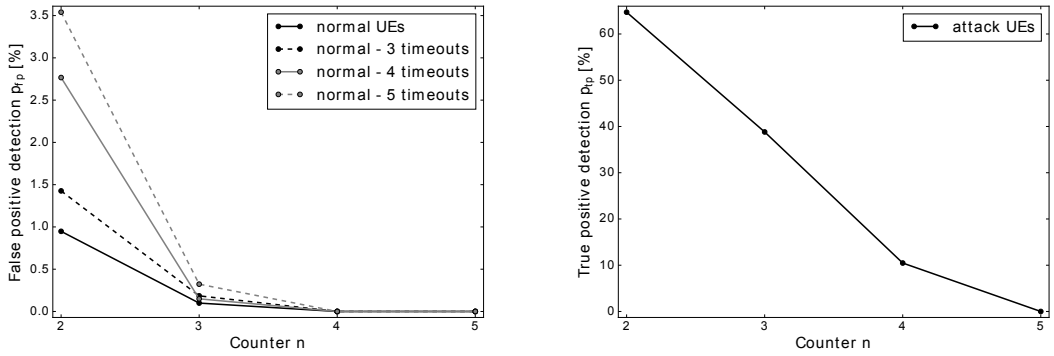
Regarding the RRC states in UMTS, described in 3.1, let us consider that a UE is *active* if it is allocated either FACH or DCH channel and *inactive* if it is in PCH or Idle state. For the evaluation of the detector, the performance metrics of interest include:

- *probability of true negative detection* p_{tn} - the portion of normal active duration which the detector classifies as normal activity;
- *probability of false positive detection* p_{fp} - the portion of normal active duration which the detector classifies as attack. This metric is also called *probability of false alarm*;

- *probability of true positive detection* p_{tp} - the portion of attacking active duration which the detector classifies as attack. This metric is also called *probability of correct detection*;
- *probability of false negative detection* p_{fn} - the portion of attacking active duration which the detector classifies as normal.

Of course, it is valid that $p_{tn} + p_{fp} = 1$ and $p_{tp} + p_{fn}$=1. The p_{fp} metric is particularly important because it shows the error the detector makes because of misclassifying normal data transmissions. In order to protect normal mobile users from being "punished" it is important to keep p_{fp} at a small value.

Fig. 5.1 shows the probabilities of false positive and true positive detection. In calculating the p_{fp} metric, we selected that some normal terminals have activity patterns which are similar to attack ones i.e., they would successively timeout 3, 4 or 5 times, so we can compare them to the attacked terminals. In practice, for example, this type of traffic could resemble machine-to-machine communication, or background control communication of an application. Results show that terminals which use more repetitive traffic trigger higher number of false detections. Moreover, as expected, p_{fp} drops with the increase of n. The true positive probability, shown on Fig. 5.1b, decreases with the increase of n. For larger n only the more persistent attackers are detected.



(a) (b)

Figure 5.1: Probability of (a) false positive and (b) true positive detection for the counter-based detector.

The counter threshold n can be used as a control, to adjust the detector's rigorousness in

different situations, as small n will improve attack mitigation (such that p_{tp} is maximal) and large n will improve attack detection (such that p_{fp} is minimal). For example, in a congested network, it's better to choose a smaller value for n such that mitigation will lower the load more efficiently, although p_{fp}-portion of normal UEs will also be punished. Contrary, in a regularly working un-congested network, large n will enable to protect normal UEs and only detect persistent attackers. It is also useful to look at the Receiver Operating Curve (ROC), depicted on Fig. 5.2 as it gives the connection between p_{tp} and p_{fp} for different n.

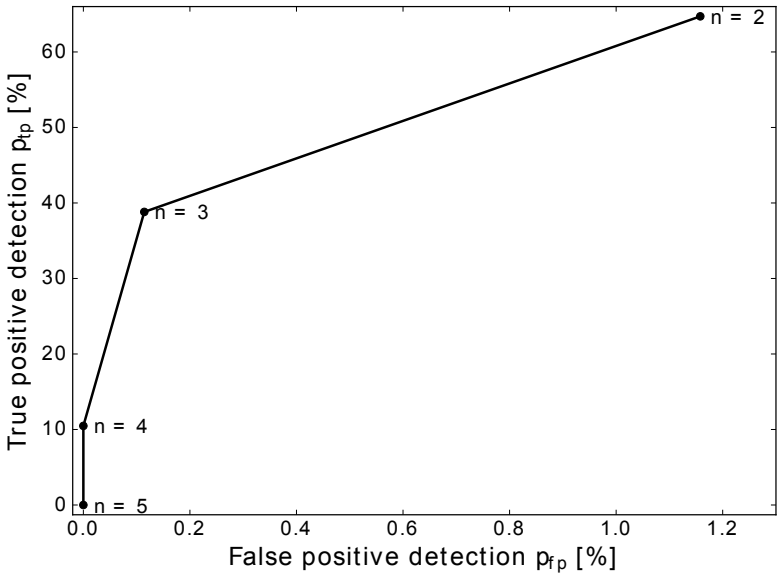


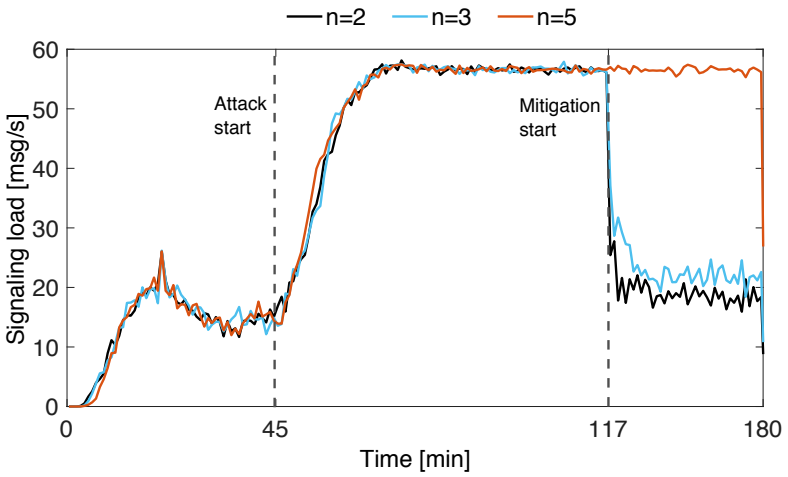
Figure 5.2: ROC curve for the counter-based detector.

In the following, we will use the counter detector in combination with a mitigation technique which bans detected attackers from communication for a short period of time.

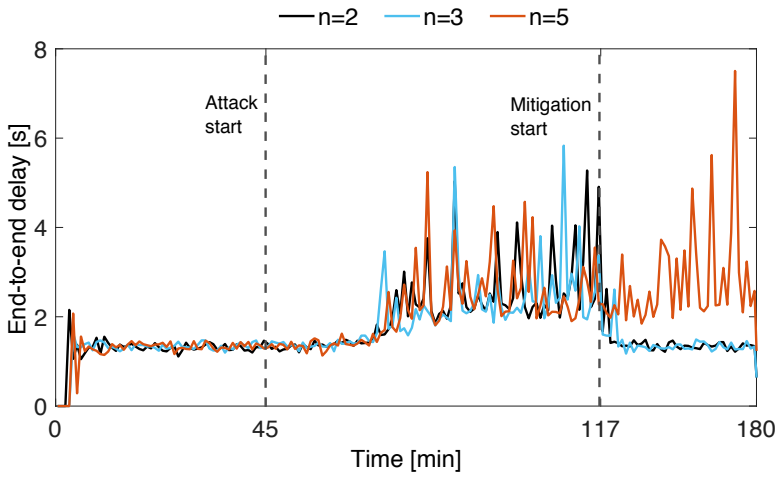
5.1.3 Simulation: detection and mitigation

To mitigate the attack we will use blocking of the attacking terminals for a time duration of $t_b = 60$s, which is applied in the moment of attack detection. The simulation setup for this purpose is same as described in Section 5.1.1 with simulated duration of 180 minutes, attacks starting gradually at 45-th minute and mitigation switched on at the

117-th minute. With this setup, we want to see the effect of the attack and the effect of detection/mitigation in the time domain. Two performance metrics of interest are shown on Fig. 5.3: the *signalling server load*, in messages per second, and the average *end-to-end delay* experienced by normal mobile users. From the moment of start of the attack, the load on the RNC is constantly increasing and after it reaches some maximum value the normal users start experiencing communication delays. Starting the mitigation with $n \in \{2, 3\}$ helps in stabilising both the network load and the experienced delay. The variation in the delay in congested system is due to the TCP retransmission of packets.



(a)

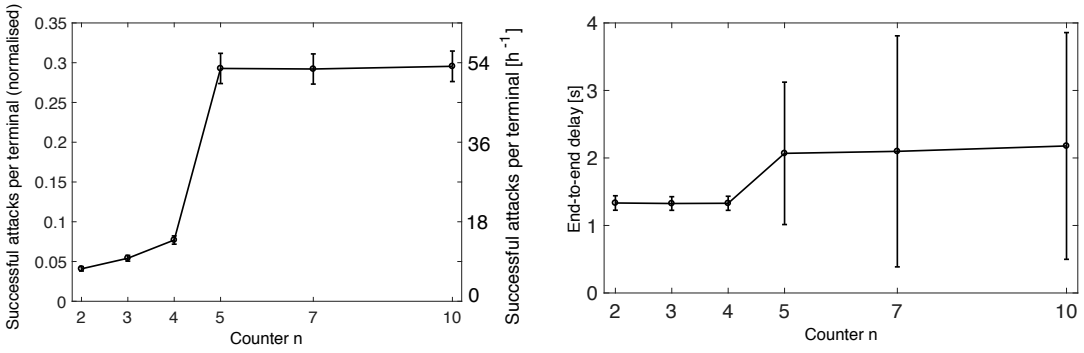


(b)

Figure 5.3: The counter-based detector in time domain. Performance metrics: (a) signalling load and (b) end-to-end delay.

The figures 5.4 and 5.5 show results of the system in steady state. The simulation setup

is as described in Section 5.1.1, with one difference: attacks and mitigation are active through the entire duration of the simulation. Fig. 5.4a depicts the average number of *successful bandwidth allocations*, or successful attacks, per malicious mobile terminal, per hour, and its normalised values. As expected, the number of attack allocations increases with the increase of n because the detector waits for more repetitions to happen before making a decision. For $n \geq 5$ our mechanism shows unsatisfying results. Fig. 5.4b shows the effect of the attack on the average end-to-end delay experienced by normal terminals. This is the delay measured on the application layer of the protocol stack. The selected number of attackers (20% of all terminals) manages to perform a successful signalling attack and overload the network which results in higher delays for the normal terminals. Results suggest that using the proposed mechanism with a threshold of $n \in \{2, 3, 4\}$, the system is kept stable and normal delays are experienced. Again for $n \geq 5$, the mechanism does not manage to mitigate the attack. The abrupt increase between $n = 4$ and $n = 5$ is due to the type of model of normal web traffic. Furthermore, the delay variability for $n \geq 5$ is much higher than for $n < 5$. This could be a result of packet retransmission using the TCP protocol in a congested system.



(a) (b)

Figure 5.4: The counter-based detector in steady state. Performance metrics: (a) successful attacks per malicious terminal (95% confidence interval), (b) end-to-end delay per normal terminal (99% confidence interval used).

Finally, we are interested in the amount of communication resources consumed by malicious mobile terminals. The analysis in the previous chapters showed that for mobile networks with dedicated approach for bandwidth allocation, such as early versions of 3G UMTS, bandwidth wastage is another negative consequence, on top of the excessive signalling. Note that in DCH state of UMTS, a bandwidth allowing high-speed transmission

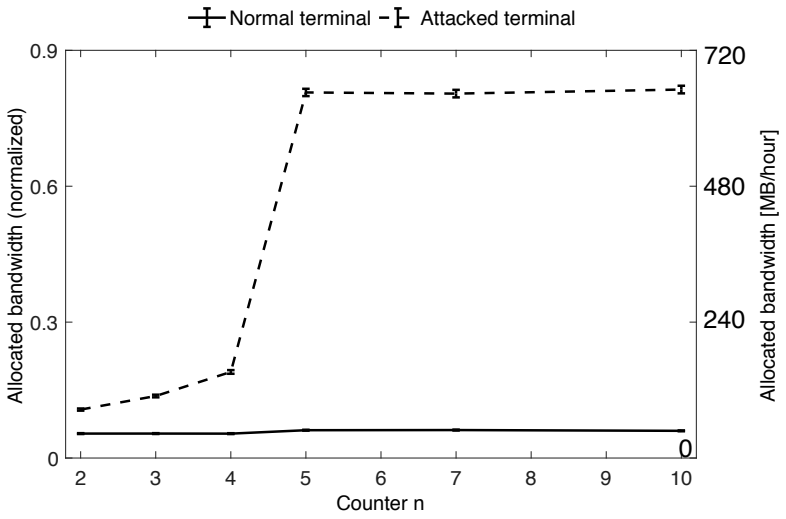


Figure 5.5: Allocated uplink bandwidth per normal and attacked terminal (95% confidence interval used).

is dedicated exclusively to the requesting terminal, a feature that is excluded in the following generations of mobile networks, like HSPA and LTE. Fig. 5.5 shows the average allocated bandwidth in uplink direction in DCH channels for attacked and normal mobile terminals in duration of one hour, and the corresponding normalised values. Results show that the amount of resources allocated per attacked terminal is significantly higher than per normal one, such that for $n = 10$ the attackers are allocated around 600 MB more than normal users in a single hour. Looking at this from a billing perspective, these users could be charged much more than usual, depending on the MNO's charging strategy. Anyway, for $n \in \{2, 3, 4\}$ the proposed mechanism manages to lower the amount of ill-consumed resources to 40-90 MB per terminal per hour. The results for bandwidth usage in the downlink direction are analog to the uplink and are not presented in this scope.

5.2 Bandwidth usage-based detector

Previous work in [61] has shown that signalling storms can be identified not only by their repetitive pattern but also by their low usage of communication resources in order to evade getting detected by traditional flooding security mechanisms. In this Section, this characteristic is used to develop a simple detection mechanism that is capable of identifying malicious behaviour in real-time. Similar to the requirements of the counter-

based detector, this detector should also represent a lightweight mechanism in terms of memory and processing power when implemented on the mobile terminal side.

5.2.1 Detector description

There are two input parameters that the mechanism needs. The first parameter is the total time that the terminal spends in a 'high' state within a given time window t_w. This is a state where communication resources are granted to the mobile terminal and is equivalent to states DCH or FACH in UMTS. Respectively the time spent in each state is denoted with t_D, and t_F. The second parameter is the time which the mobile terminal spends in 'high' state but does not transfer any data (stays idle), also in a time window t_w, which is denoted with t_{Di} and t_{Fi} for the respective states in UMTS. The detector works by calculating the ratio $\frac{t_{Fi}+t_{Di}}{t_F+t_D}$ whenever resources are (de)allocated i.e., for every state change. Then the Exponential Weighted Moving Average (EWMA) algorithm is used to calculate the cost function C as:

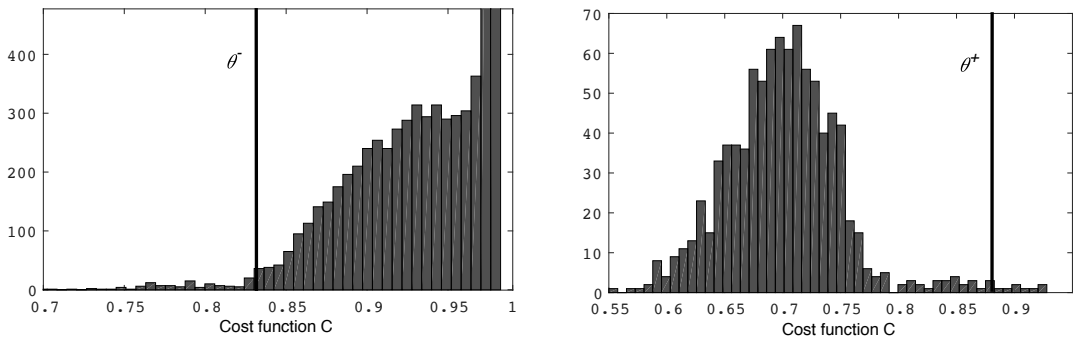
$$C[k] = \alpha \frac{t_{Fi}[k] + t_{Di}[k]}{t_F[k] + t_D[k]} + (1 - \alpha)C[k-1], \tag{5.1}$$

where $k \in \mathbb{N} > 0$ is the index of the state change, $0 \leq \alpha \leq 1$ is a weight parameter and $C[0] = \frac{t_{Fi}[0]+t_{Di}[0]}{t_F[0]+t_D[0]}$ is the initial cost value. This cost function enables detection of attacks on both FACH and DCH channels in UMTS, and with suitable adjustments it could be adapted to any network with similar functionality, like LTE. The calculation of the cost function is foreseen to run as a background process in each mobile terminal or in a centralised network node that has the needed information for all terminals, such as the RNC in UMTS and the eNodeB in LTE.

The decision making is based on the value of the cost function. As defined, C is between 0 and 1 with values closer to 1 indicating higher probability of an attack. To define when an attack is detected let us suppose that the mechanism is running for long enough time so that the average value of the cost function C_{avg}, observed since the mobile activation, is stable with minor variations. Then, a malicious behaviour is detected if $C \geq \beta C_{avg}$ where β is a value close to, but larger than 1. Our simulations have shown that a suitable choice

would be $\beta = 1.5$ meaning that 50% increase of C above its average value is an indicator of attack. Note that C is calculated within a time window t_w while C_{avg} is calculated from the activation of the mobile terminal.

There are two problems with this type of decision making. First, if $C_{avg} > 1/\beta$, an attack cannot be detected. This could happen if an attack is ongoing from the moment of activation of the mobile terminal, thus producing high C_{avg} value such that $\beta C_{avg} > 1$. The second problem appears for heavy traffic users, for example users who use video streaming or voice communication. These terminals use a big portion of the granted resources and therefore have very low C_{avg}. In this case the product βC_{avg} is still very small and normal usage is often clarified as attack.



(a) (b)

Figure 5.6: Threshold setup on SECSIM simulated data for (a) p_{fn}=0.01 and (b) p_{fp}=0.01.

To address these issues we propose setting up two thresholds for C: an *upper threshold* θ^+ above which we make a decision of an attack, and a *lower threshold* θ^- below which we make a decision of normal behaviour. Both decisions are irrespective of C_{avg}. Setting up these thresholds should be based on offline traffic analysis by the mobile operators. A frequently used thresholding technique is one based on a fixed value for the probability of false positives p_{fp}, which is defined as the fraction of time in which an attack is detected but not existing. Similarly, the *false negatives probability* p_{fn} is the fraction of time in which an attack is ongoing but is not detected. Having the probability distribution of offline measured C values for bandwidth requests classified as normal, we set θ^+ as the threshold above which statistically 1% of normal requests will be declared as attack. Similarly, θ^- is set up for a 1% fixed probability of false negatives p_{fn} based only on

offline-measured C values of bandwidth requests classified as attack. Fig. 5.6 shows the threshold setup on SECSIM generated data, while Fig. 5.7 shows an example of C in time and the setup thresholds. The proposed two-threshold decision making approach solves the above mentioned problems. Furthermore, for $\theta^- < C < \theta^+$, an attack could still be detected by checking if $C > \beta C_{avg}$.

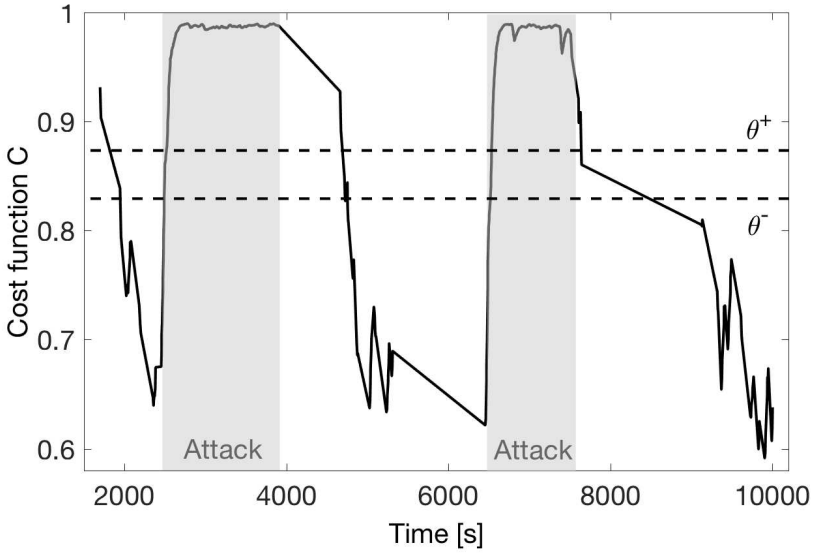


Figure 5.7: An example of the cost function C in time, with $\theta^+ = 0.88$ and $\theta^- = 0.83$, and two attack intervals.

The proposed mechanism allows detection of signalling attack behaviour in a real-time manner. It works with calculating the cost function $C[k]$ for each data transmission at instance k. In any case, single data transmissions are not classified as attack/normal but rather a decision is made upon a group of data transmissions in a time window t_w. This is because single attack transmissions or attacks with low frequency cannot form a signalling attack and cause disruptions to the network. The mechanism works in real-time because the observation window slides through time and the cost function is calculated using the EWMA averaging.

5.2.2 Detector evaluation

To evaluate the detector, we implemented its functionalities in the SECSIM simulator, and we setup a simulation as described in 5.1.1 with 500 mobile terminals out of which

150 are attacked and $\alpha \in \{0.01, 0.1, 0.3, 0.5, 0.7, 1\}$. The simulation is repeated five times for each α value with different seeds for the random number generators and results are averaged over all runs. The rest of the parameters are configured as follows: $t_w = 60$s, $\beta = 1.5$, $\theta^- = 0.83$ and $\theta^+ = 0.88$. Attacks happen by malicious terminals at a random time of the simulation and in intervals of random length. Note that the ratio of attacked / total terminals does not influence the evaluation of the detector.

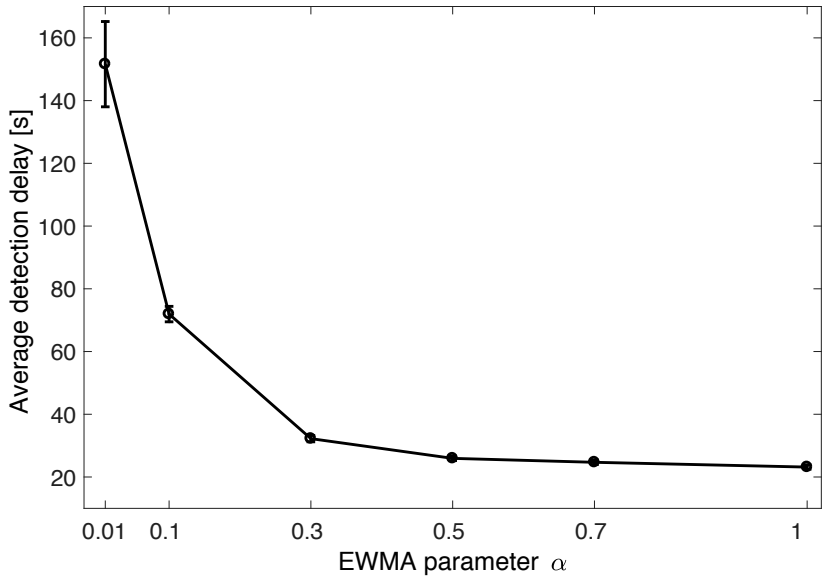
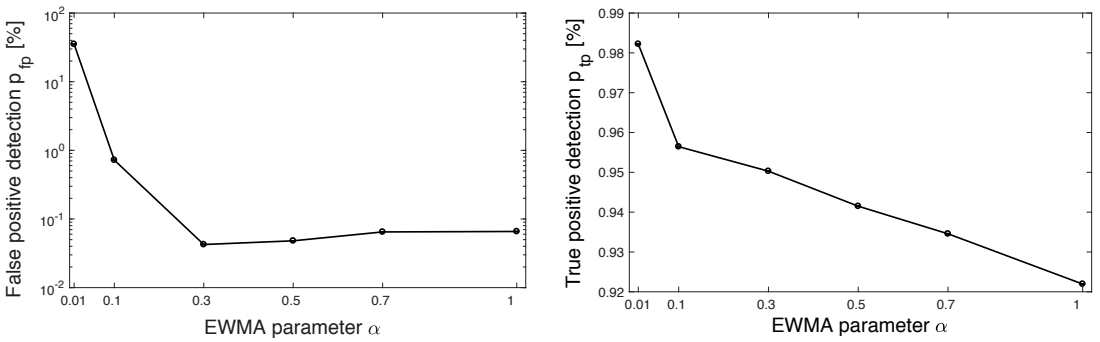


Figure 5.8: Average detection delay for the bandwidth usage-based detector (95% confidence interval used).

One performance metric of interest is the *detection delay* τ i.e., the delay between the time instant of attack start and its detection, and is a function of the moving average parameter α and the width of the sliding window t_w. Fig. 5.8 shows the average detection delay for different values of α. A smaller value for α in the calculation of the cost function gives more importance to historical than present C values and makes the detection more rigid which is shown by the high values of the detection delay. Increasing α makes the detector more flexible and improves the detection delay. However, for higher values of α we expect the cost function to change too rapidly and increase the number of false detections, a hypothesis which will be tested in the rest of this Section.

Fig. 5.9a, shows the probability of false positive detection p_{fp} as a function of the EWMA parameter α. Lower α values not only slow down the detection but also result in high



(a) (b)

Figure 5.9: Probability of (a) false positive and (b) true positive detection for the bandwidth usage-based detector.

p_{fp} values. As α increases close to 1, p_{fp} also increases, because the detector makes faster decisions, which may also be less accurate. The selection for $\alpha = 0.3$ seems most appropriate as it minimises the false positives probability, to a value of 0.04%. Fig. 5.9b depicts the portion of detected attack intervals among all malicious behaviour. Smaller values of α produce higher p_{tp} values because the detector waits for a longer time interval before making a decision. Note that in our experiments during an attack, the attack traffic is mixed with the traffic of normal web browsing application. This causes the p_{tp} to have lower values as some parts of the attack interval will still be identified as normal. In case of a deliberate attack with only an attack application installed on mobile devices (without normal web browsing traffic in the background), the detector's performance would improve.

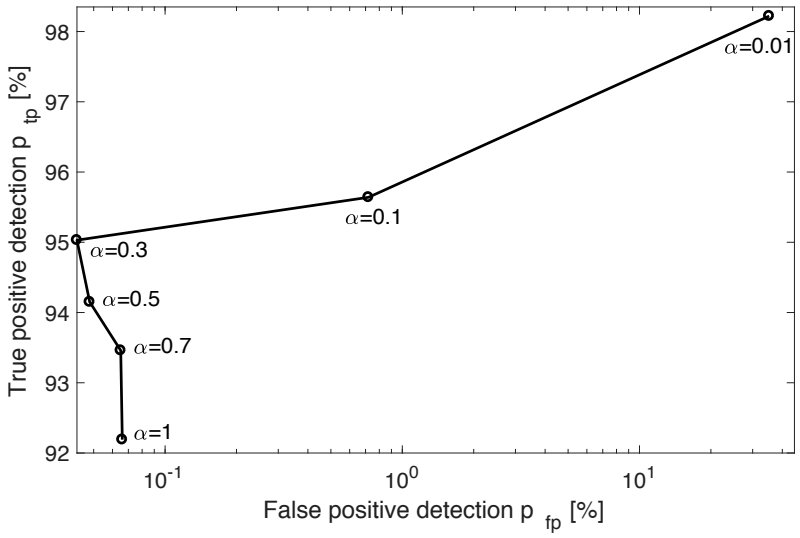
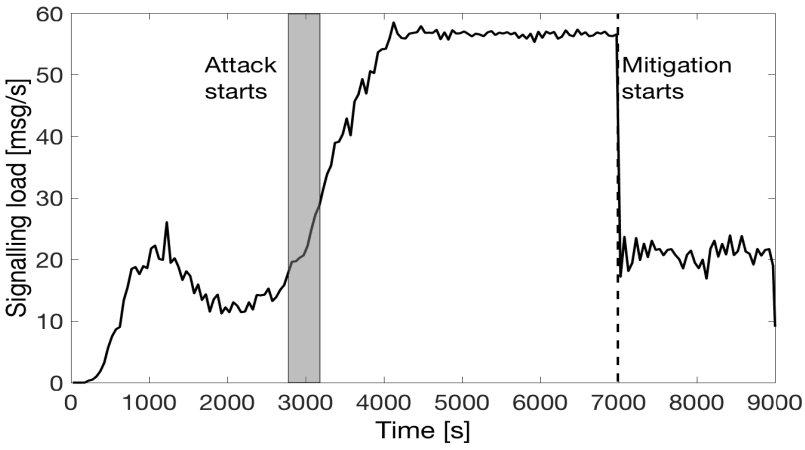
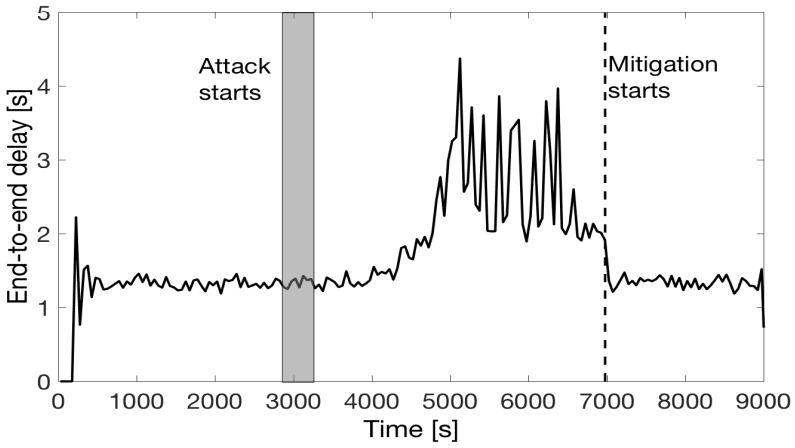


Figure 5.10: ROC curve of the bandwidth usage-based detector.



(a)



(b)

Figure 5.11: The bandwidth usage-based detector in time domain. Performance metrics: (a) signalling load and (b) end-to-end delay.

Finally, Fig. 5.10 combines the p_{fp} and p_{tp} metrics into the ROC curve for the proposed detector. Values in the top-left corner of the graph are most desirable, as it produces the highest true positive and lowest false positive detection probabilities. The simulation results suggest that $\alpha = 0.3$ is the most suitable value, producing 95% true positive and 0.04% false positive detection.

5.2.3 Simulation: detection and mitigation

In this part we use both the detector and the mitigator at the same time, and look at the attack scenario in the time domain. Same as in 5.1.3, we will use a blocking time with duration of $t_b = 60$ s, which is done immediately when attack is detected. We set

the simulated duration to 180 minutes, with attacks starting gradually at 45-th minute and mitigation switched on at 117-th minute. The metrics of interest are the *load on the signalling server* in the RNC in terms of processed messages per second and *end-to-end delay* measured on application layer experienced by normal users. The results shown on Fig. 5.11 show that in the period without attacks (0 - 45 min) the network load has a small peak because of the activation of mobile terminals (which includes exchange of signalling messages with the RNC), after which the load stabilises. The end-to-end delay is also stable. In the period when the attack is ongoing (45 -117 min) the load on the RNC starts to increase until a certain point when it reaches a maximum value. At this point the buffers in the signalling server are congested which results in higher delays for the normal users. Using the proposed detector with the blocking mitigation technique from 117-th minute manages to decrease the load in the network and stabilise the experienced delay.

5.3 Chapter summary

The proposed detectors are lightweight mechanisms for real-time detection of signalling storms. Due to their low memory and processing demands, they could easily be implemented on the mobile terminal side, although they could also operate on network side. The counter detector is based on counting successive bandwidth allocations of same type and setting a threshold of the number of legitimate ones. The bandwidth detector is based on the fact that malware applications causing signalling attacks send small portions of data to evade being detected by flooding detection mechanisms. The proposed cost function uses this fact to map the terminal's behaviour into a value between 0 and 1, with the two extremes respectively indicating completely normal and malicious behaviour, and set thresholds according to which the detection decisions are made. Both detectors were evaluated using the probabilities of false positive and true positive detection, and the corresponding receiver operating curve. The lessons learned in this Chapter are as follows:

- A detection mechanism based on a simple counter of consecutive same channel allocations does not provide satisfactory results, with a true positive probability as low as 40%.

- Simulation results provided for joint work of the bandwidth usage based detector and a mitigator show that it is capable of maintaining the network's stability and helping in providing good quality of service to the normal users, deducted from a true positive probability of 95% and false positive probability of 0.04$.
- The counter based detector, combined with a mitigator, manages to lower the network load, but at the expense of increased probability of false detection i.e., at the expense of blocking normal calls which are detected as attacking.
- An online, simple mechanism with low complexity, low memory and processing demands is capable of solving the problem with signalling storms. Such solutions could be further improved once implemented in a working mobile network.

To further improve its performance, the counter detector can be upgraded with additional mechanisms like deep IP packet scans of suspicious terminals. The bandwidth usage-based detector could make further improvements by dynamically adapting the parameters of the detector (α, θ^+ and θ^-) to the type of the communication of the mobile terminal (human generated, machine to machine communication, sensor data communication, etc.).

Chapter 6

Conclusions and future work

6.1 Conclusions

This thesis described the research done in the field of performance analysis of mobile networks under signalling storms. Signalling storms / attacks are a novel type of denial of service attacks on the control plane of the network. They could be triggered by specially designed malware for smart devices, or as a byproduct of poorly designed legitimate applications. Negative consequences on operating 3G UMTS networks have been documented as either: service degradations, partial, or full network outages. Our research, using mathematical modelling and simulation techniques, analysed the impact of the attack on both 3G and post-3G networks. It questioned if the network can defend its stability by adapting some of its internal parameters, what are the bottlenecks in the network from architectural perspective, what is the influence on the data plane and normal mobile users, etc. Further on, the thesis proposed two detection techniques and evaluated them in a simulation environment. The following describes in more details the conclusions made through the thesis.

Chapter 2 categorised threats and corresponding countermeasures in mobile networks, listed the most common types of malware, introduced the signalling storms and described the analytical and simulation frameworks used in this work. This Chapter contributed in having a more complete view of the security aspects of mobile networks and in identifying

the similarities between different types of attacks.

In Chapter 3 the vulnerability in the RRC mechanism in UMTS was described and its stochastic model from a mobile terminal side was proposed. The model was used to analyse if the network can use some of its internal parameters to defend against signalling storms, and a conclusion was made that the inactivity timers and call setup delays can be used to lower the impact of the attack. Some general recommendations on the choice of these parameters were also given. Furthermore, to quantify the influence of the inactivity timers, we proposed a network-side model using tools from queueing theory. It used two classes of calls traversing an open network model and a limited number of data channels, alike in UMTS Rel '99 version. The model allowed us to conclude that: (i) the bottlenecks in the network are the signalling servers, and base stations, depending on their respective capacities, the number and malicious devices under control and their attack rate, (ii) the limitation of dedicated data channels indirectly acts as a self-defensive mechanisms in signalling-related attacks and (iii) the inactivity timers cannot prevent from network outages in signalling storms, but can decrease their impact. Regarding point (iii) we proposed a *dynamic timer* which sets the inactivity timer as a function of the load in the network. This approach manages to lower the load in the network under attack, but at the expense of the number of served normal calls. This means that the inactivity timer is not just a trade-off between the effective bandwidth usage and number of connections, but also a trade-off between the signalling load in the network and the number of unserviced normal calls.

In Chapter 4 we extended the network model proposed earlier such that it represents different network technologies, preceding and following 3G. To facilitate the analysis, we distinguished two groups of 3GPP mobile technologies: *traditional* networks that incorporate mobile data services using concepts from the circuit-switching domain (using dedicated resource allocation), and *contemporary* networks which allow bandwidth reuse through sharing of resources. Numerical analysis of the system led to the following conclusions. First, the shift in the design of the radio access system, from using dedicated channels to shared ones, has made mobile networks more resistant to signalling storms. Nevertheless, 4G/LTE networks are still vulnerable to signalling storms. More importantly, they

are more susceptible to the single-point-of-failure problem due to LTE's non-hierarchical flat topology in which the main signalling server, located at the core network, is directly connected to the base stations with no intermediate servers at the access network to protect it if some of the massive number of mobiles under its control misbehave. Second, increasing the signalling capacity is not an effective long-term solution, because signalling traffic is expected to outpace the exponentially growing mobile data traffic, with future developments in areas like M2M and IoT. Therefore, active and real-time detection and mitigation systems are needed to protect future generation networks.

Finally, based on conclusions in our earlier work, we proposed two real-time, detection mechanisms in Chapter 5, which were implemented in the SECSIM simulator on the side of the mobile terminals. The counter detector is based on counting successive bandwidth allocations of same type and setting a threshold of the number of legitimate ones, while the bandwidth detector is based on the fact that malware applications causing signalling attacks don't send any data or send small data portions. Both detectors were evaluated using the probabilities of false positive and true positive detection, which showed satisfying results: most importantly producing low false positive probability. Further simulation analysis for joint work of the detectors and a simple mitigation technique showed that both mechanisms are capable of maintaining the network's stability and helping in providing good quality of service to the normal users. Therefore, simple detection mechanisms, like the ones proposed, could be effectively used in the fight against signalling storms.

6.2 Future work

In this Section we discuss the areas where our work can be extended, and propose possible future research directions.

In the mobile terminal model in Chapter 3 we defined a cost function based on the probabilities of normal and attack states. Defined this way, the cost function gives a measure of the attack's impact, regarding channel occupation time by normal and malicious calls. Conclusions of this research indicated that duration of malicious channel occupation is not the only negative aspect of signalling storms, but also the rate of it happening. Therefore,

we could further improve this part by including the influence of state transition probabilities in the cost function. The work in this Chapter showed that the inactivity timers, which are usually set to a fixed value by the MNOs, can play an important role in the defence against signalling storms. While these timers cannot fully alleviate the attack, if adjusted properly depending on the network load, can serve as a control to lower the effect of the attack. Therefore, we consider that future mobile networks should employ configurable, if not fully automated, inactivity timers which would give MNOs more freedom in controlling their networks.

In Chapter 4 we studied traditional and contemporary networks under signalling storms using a queueing network model. Within Section 4.4.2 we first compared two candidates of both network groups: UMTS Rel 99, and LTE Rel 8. For the next step in this direction, we could first examine the storms influence on LTE Advanced networks, and then also incorporate 5G networks in the model. We consider that the data plane part of the 5G model could be used as a new module in our global network model, but any needed changes in the control plane would also require modifications of the global model too. In the same Chapter, we examined the storms' influence on different traffic types: web browsing, video streaming and machine-to-machine. Further on, we could improve this part by using more realistic, mixed-traffic models, which will combine the different types together on a network level, and afterwards on a terminal level. Finally this Chapter compared the mathematical model of UMTS with the simulation model implemented in SECSIM. The preliminary results shown in this part did not show a good fit between the two models, regarding Fig. 4.12. This is mostly because the simulation uses infinite bandwidth capacity without the possibility of call blocking, while the mathematical model uses a fixed capacity and number of dedicated channels, and incoming calls are blocked in case all channels are occupied. Our future work is to implement bandwidth capacities in the simulation, which will further require modifications in the allocation of shared channels, and possible changes in the frequency reuse planning when simulating multiple network cells.

Lastly, in Chapter 5 we proposed two signalling storm detection mechanisms, which were implemented and evaluated in the SECSIM simulator. While both detectors showed sat-

isfactory results, they could still be improved. The bandwidth usage-based detector could further be improved by adjusting its parameters to the type of communication of the terminal (human generated - also taking into account the most frequently used traffic types by each user, machine-to-machine, sensor reports, etc.). The task for identification of the traffic type for each terminal would require applying techniques from Machine Learning (ML), while the devices generating sensor/machine data should be available in a database. The counter-based detector could be improved by selecting IP packets for deeper inspection, although it could also be combined with the bandwidth usage-based detector, forming an ensemble classifier. An important consideration here must be given to attack detection within future communication types, such as data automatically exchanged by machines.

6.3 Future projects

The research area of attack detection in this field is probably the most important area, as the signalling problem still exists, and as our analyses showed - it will continue to cause problems in future network generations. Our future projects will focus on techniques from the Machine Learning (ML) field, which is selected as a natural switch from the stochastic modelling area and Markov processes. Machine learning is a sub-field of Artificial Intelligence (AI) that evolved from *pattern recognition* and *computational learning theory*. This area is also very popular in the last few years, both in research and industry. Some of the ML techniques that could be considered in this research include: neural networks, random neural networks and deep learning (eg. TensorFlow):

- **Neural Networks (NN)** - represent computer programs that operate in a similar manner to the human brain, with objective to perform cognitive functions as problem solving [112]. They use networks of interconnected nodes, called *neurons* that exchange impulses between each other via synapses (connections). Neurons are organised in layers which could be input, hidden or output layers. Synapses store weights, which are used in the computations of data. The learning ability of NN comes from mathematical optimisation and is based on finding a optimal function f^* in a class of functions F. This is done by defining a cost function

$C : F \to \mathbb{R}$, such that the optimal function has the least cost among all functions $C(f^*) \leqslant C(f), \forall f \in F$. There are three types of learning paradigms: supervised, unsupervised and reinforcement.

- **Random Neural Network (RNN)** - networks that build up on top of NN adding positive (excitatory) and negative (inhibitory) spikes between nodes, or from the outside to the nodes [113]. In this way RNN closely represents the signals transmitted in a biological neural network. A potential of each node can be defined as the non-negative sum of positive spikes received, and each node can fire a spike when its potential is positive. RNN's general model consists of connections between all nodes, so the model represents a recurrent neural network. The product form solution of the model is presented in [113], while some learning aspects and possible applications are presented in [114] and [115] respectively.

- **TensorFlow** - is an open source software library for machine learning developed by Google Research [116]. Another popular direction of ML is downlink (DL) networks, or Deep Neural Networks (DNN), which represent NN with multiple hidden layers [117]. Nowadays, NN and DL are widely used in solving machine learning problems from big sets of data. It is a successor of DistBelief system, and consists of multiple algorithms for deep learning neural networks suitable for use in many areas and on multiple computational platforms. TensorFlow is suitable for fast implementation on various problems and trying out research ideas, and as such is selected for potential future research in our field.

Besides these listed techniques, we could also examine other algorithms for anomaly detection and unsupervised learning. The selected technique should initially learn from data which would be generated with SECSIM, due to lack of data from real networks. Finally, we could modify the proposed detection mechanisms to work on detection of other DoS attacks, such as: SMS floods, premium SMS, command and control behaviour, compromised femtocells and others.

Appendix A

SECSIM remote control

The Mobile Networks Security Simulator (SECSIM), described in Section 2.3.2, played an important part of the research done for this report. It was used to develop and test signalling attacks detection and mitigation techniques, and to test many research ideas within and out of the scope of this thesis. This section presents some the technical work regarding upgrading SECSIM's functionalities.

During the work with SECSIM, as part of an international research project, there has been a need to run simulations from a remote location. Moreover, simulations needed to be run in an automated way, such that a machine can remotely start simulations and retrieve results for analysis, in an encrypted communication. To enable this, some of the functionalities that should be implemented were: simulation configuration should be simplified using only the most basic parameters, communication should be done using a client-server approach, authentication of hosts should be provided, simulation data should be organised and stored in a database, multiple parallel simulations should be able to run at the same time and data retrieval should be done using the File Transfer Protocol (FTP) with encryption. The architecture of the provided solution is described on Fig. A.1.

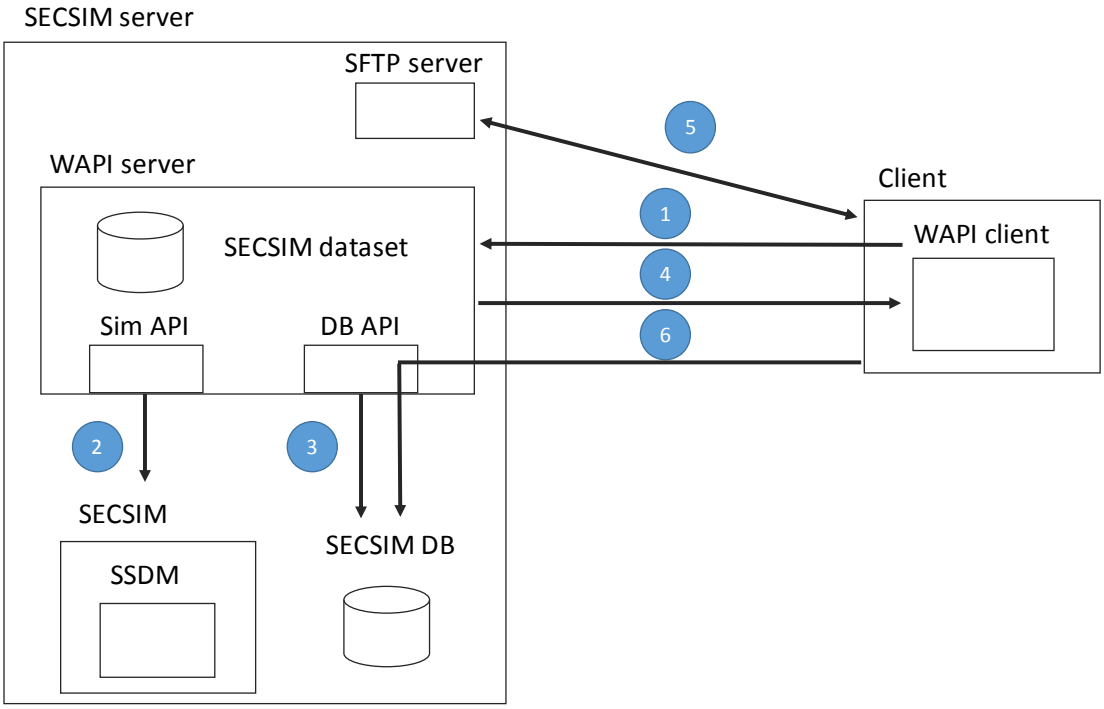


Figure A.1: The architecture enabling remote control of SECSIM.

The provided solution wraps up the simulator in the SECSIM server. The SECSIM server further contains the WAPI server which is the central point in the architecture. WAPI is a shortcut for WOMBAT API, the Application Program Interface for a tool developed in the WOMBAT FP7 project [118], that enables encrypted and automated remote transfer of data. The data is organised in a dataset called SECSIM dataset, which also provides the client methods for data manipulation. Access to the data is asynchronous, provided by its engine built in Twisted Python, and data transfer is encrypted using asymmetrical encryption techniques. The WAPI server further contains two interfaces: Sim API for communication with the SECSIM simulator, and DB API for communication with a database. All of the WAPI server functionalities are developed in Python 2.7 programming language [119]. The SECSIM simulator has a package called Signaling Storm Detector and Mitigator (SSDM), as depicted in the figure, which contains the developed detection and mitigation techniques described in Chapter 5. The SECSIM DB is a NoSQL database with document-oriented structure, developed in MongoDB [120], and is responsible for storage of the simulated data, and simulation configurations requested by the client. Finally the server side contains an Secure File Transfer Protocol (SFTP) server which is responsible

for transfer of big data files from the server to the client. In this scope, it is important to note that all of the building blocks on the SECSIM server side were implemented on a single machine running Kubuntu, version 14.04 LTS. A brief description of the workflow of the provided solution is provided with the following steps, in correspondence with Fig. A.1:

1. The client connects to the server using `https://<ip_address>:<port>/secsim` url, and asks for a simulation using the `request` method implemented in the WAPI server, providing the parameters needed to configure the simulation.
2. The WAPI server checks if it already has the data for the requested simulation in the database, and if not, it starts a new one.
3. When simulation is done, the WAPI server parses the textual result files and saves the data in the database.
4. The WAPI server tells the WAPI client that data is available, sending the simulation identifier `simId` as a parameter.
5. The client authenticates with the SFTP server using a username/password or a client certificate.
6. The client retrieves the data from the database through the WAPI server.

The WAPI server is responsible for most of the functionalities in the system, and their description is given in the following. Firstly, a brief introduction of the WAPI tool is provided, after which the building blocks of the system are explained in more details.

A.1 WAPI server

The WAPI server represents the focal point of the architecture enabling remote access to SECSIM. The tasks it is responsible for are: to organise the simulated data in a dataset *SECSIM dataset*, provide methods to the WAPI client to query the dataset, to configure and start simulations, to parse simulation results and save them in a database.

A.1.1 WOMBAT API description

One of the key goals of the WOMBAT Project [118] consisted in the development of mechanisms for the integration and sharing of data generated by the various security related data feeds developed and maintained throughout the project. The WOMBAT partners proposed to define a common API, called the WAPI, to be shared among all participants in order to address the above issues and simplify the task of the data consumer willing to take advantage of these datasets. The WAPI is a remote API based on Simple Object Access Protocol (SOAP) that allows data consumers to retrieve remote information from sources according to a given communication protocol. WAPI facilitates the integration of information generated by multiple data feeds and enables analysts to write programs that combine data from several information sources through a uniform set of primitives. WAPI decouples the structure and the characteristics of each dataset from the clients: dataset maintainers can decide what they are eager to share, in which format and to whom, and can dynamically refine their dataset structure (e.g. add new information types to the existing datasets) without any need to update the querying clients that discover the dataset modifications at runtime.

A.1.2 SECSIM dataset

The organisation of SECSIM dataset is depicted on Fig. A.2. It contains the following objects, which are exposed to the client: `SECSIM`, `simulation`, `scalar`, `vector`, `statistic` and `simConfig`. The `SECSIM` object is the object the client gets automatically when connected to the server. Using this object, the client can use the method `listSimulations` to get a list of simulation identifiers which are stored in the database, and the method `request` with arguments `inputParams` to start a new simulation. Although one of the advantages of the simulator is the high configurability, in the scope of this tool there was a need to simplify the remote configuration of simulations. Therefore, we have selected a list of the 24 most basic parameters `inputParams` which are exposed for remote access. A list of these parameters is given in Table A.1. Finally the SECSIM objects has a method `getSimulation` with parameter `simId` which returns a reference object to the requested

simulation.

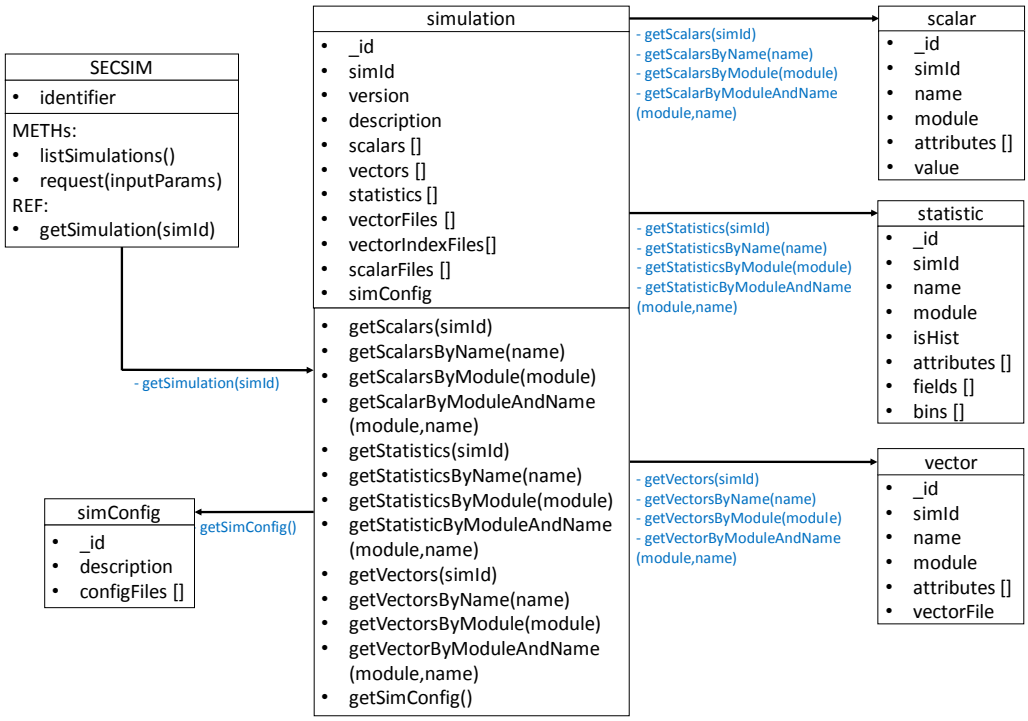


Figure A.2: SECSIM WAPI dataset description.

The object `simulation` is used to merge all the different types of data belonging to a simulation in a single object. The three main types of objects storing data are: `scalar`, `vector` and `statistic`. The `scalar` object stores results as single numbers, the `vector` stores results in arrays of time-series values and the `statistic` object stores histograms and statistics related to a given quantity. The attributes `scalarFiles`, `vectorFiles` and `vectorIndexFiles` represent lists of `fileRef` objects (Fig. A.3) which contain links to the actual text files with simulation results. The `fileRef` and `attribute` objects on Fig. A.3 are used for internal data organisation purposes and are not exposed directly to the WAPI client.

Furthermore, the `simulation` object contains the `simId`, `version` and `description` attributes which contain the identifier, Omnet++ version and a short user description. Finally, the `simConfig` object contains the simulation configuration parameters, by providing links to the actual configuration files in the filesystem. These values are regarded as input values for the simulator. The methods that the `simulation` object provides are used

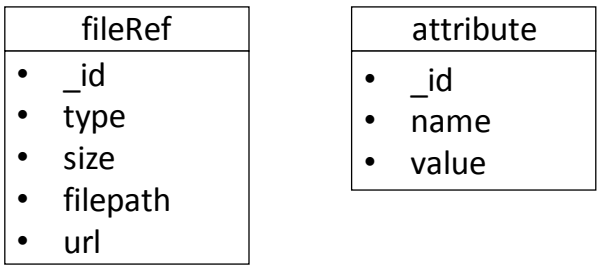


Figure A.3: Objects that are used in the SECSIM dataset but not directly exposed to the WAPI client.

for retrieval and filtering of data, such as `getScalars`, `getVectors`, etc. The names of the methods are self-explanatory and easily describe their function. The `scalar`, `vector` and `statistic` objects are the lowest-scale objects that store the actual data. Therefore they do not contain any methods.

A.1.3 Sim API

The Sim API is an interface inside the WAPI server that is responsible for interaction with the simulator. It is responsible for: configuration and running of simulations, maintaining a queue of simulation requests and maintaining the threads run in parallel. The configuration of each simulation run is done according to the parameters in the `request` method from Table A.1. It mainly works by modifying text based files used by SECSIM to configure the work of entities like UEs and the corresponding servers, while the core parts of the mobile network models in SECSIM are not configurable remotely. The client requests come to a First-in-first-out (FIFO) queue with a single server and are executed when a processor is available. The current implementation uses only a single processor for running simulations due to the available resources on the machine on which it works. Namely, a computer with two processors is used, of which one is used for the run of the WAPI server and the second for running a single simulation at a time. Finally, simulations are started in a parallel thread in order not to block the work of the WAPI server.

Table A.1: Input parameters in the request method used to configure simulations.

Parameter name	Type	Default value	Description
`scenarioName`	string	'Default'	A short descriptive name
`numUE`	int	10	Number of UEs in the network
`simTimeLimit`	int	60min	Simulated time, in minutes
`mobileUE`	boolean	False	If UEs are mobile or static
`useWebApp`	boolean	False	Web browsing app
`useRrcAttackApp`	boolean	False	RRC attack app
`useSmsApp`	boolean	False	SMS app
`useSmsSpamApp`	boolean	False	SMS spam app
`useChatApp`	boolean	False	IM app
`rrcAttackState`	string	'DCH'	RRC attack state ('DCH or 'FACH')
`numRrcAttackUE`	int	5	Number of UEs that use RRC attack application
`rrcAttackStart`	int	0min	Time instance at which an RRC attack starts, in minutes
`rrcAttackStop`	int	99999h	Time instance at which the RRC attack stops, in hours
`useSsdm`	boolean	False	If the SSDM functionality is used
`counterDetection`	boolean	False	Type 1 detection: counter based
`costDetection`	boolean	False	Type 2 detection: bandwidth usage based
`counterThreshold`	int	3	A threshold of consecutive transitions. Used if `useCounter=True`
`isMitigating`	boolean	False	If mitigation is switched On
`mitStartTime`	int	99999min	Time instance at which mitigation should be switched on, in minutes
`blockTime`	int	60s	Time duration to block an attacking UE, in seconds
`numSpamUE`	int	5	Number of spamming UEs
`interSpamTime`	int	5s	Average time between two waves of generated SMSs, in seconds
`premiumSmsProb`	float	0.01	Probability of an SMS sent to a premium number, in range [0,1]
`interChatMsgTime`	int	3min	Average time between two waves of generated IM messages, in minutes

A.1.4 DB API

The DB API is the interface that allows the WAPI server to parse simulation results from their original text format and interact with the database. This interface is needed to enable automated retrieval of data, instead of transfer of large text files. As mentioned earlier, MongoDB is used for data storage, while the code that enables interaction with the database is built in Python. MongoDB was selected as a suitable database because of its NoSQL document-oriented structure, rather than a traditional table-based relational structure. It allows us to save objects in a JavaScript Object Notation (JSON) style, which is widely used in many areas today. The DB API is used to store the three main type of objects: `scalar`, `vector` and `statistic`, as well as the additional `fileRef` and `attribute` objects. Data regarding the client requests is also saved in the database, as this information is needed in the SECSIM dataset and Sim API.[1]

A.2 SFTP server

The vector data files store time series simulation results, and on many occasions these files grow in the range of tens of gigabytes, which is a problem regarding its parsing and database storage. First, parsing of these files would take a long time on a two-core processor, and second, the uncommercial version of MongoDB does not support storage of items larger than a few gigabytes. For this purpose, the large vector text files are not parsed and saved in the database. Instead they are provided to the client via FTP transfer. Since this transfer is not supported by the WAPI server, we provided a parallel service using an SFTP server, which is selected as the most suitable solution. Furthermore, the SFTP provides additional host authentication and data encryption, which was a predefined requirement for the system. There are two possible ways of host authentication: via username/password or via digital certificates, while the data encryption is done again by asymmetric encryption techniques. It further sets the client access permissions on the server for the FTP transfer, configured using the standard tools provided by the Ubuntu

[1]The author would like to thank Mr Gökçe Görbil and Mr Antoine Husson, from the Intelligent Systems and Networks (ISN) group at Imperial College in London, for completing most of the work in DB API.

system. These files can be accessed by the client through the `fileRef` object in the WAPI server (Fig. A.3) which keeps a `url` field with the FTP link to the file.

A.3 Workflow example

The following example code shows a typical workflow for requesting a simulation execution via the WAPI interface and collecting the simulation results. The simulation has a scenario with RRC attack, as can be deduced by the parameters in the request method. The following code is run on the client side in a Python environment. Comments are marked with '#' sign at the beginning of the respective lines.

```
# make a request for a simulation
> simId = SECSIM.request({numUEs: 1000, simTimeLimit: 180, numAttackers: 300,
  isPchEnabled: True, rrcAttack: DCH, attackWaitTime: 2})
# --- simulation is queued and run when a processor is available ---

# get the simulation object, upon receiving the simId of the run simulation
> sim = SECSIM.getSimulation(simId)

# get the number of successful attacks for all UEs and calculate the mean
# number of successful attacks per attacking UE
> attacksUEs = sim.getScalarsByName(dchAttacks:count)
> meanAttacks = numpy.mean(attacksUEs[:numAttackers])

# get time series data for RRC state changes for a particular module (RRC
# module in ue[0]). Note that vector data is not stored in the database so an
# url is returned to get the data via FTP
> vec = SECSIM.getVectorByModuleAndName(GenericNet.ues[0].rrc, rrcState:vector)
> vecFile = vec.vectorFile
> ftpUrl = vecFile.url
```

A similar workflow example for running the SSDM functionality in SECSIM is presented below. Results are retrieved for a `statistic` result of UE[0]. In this case the statistic is a histogram, which can be seen by the `stat.bins` value, or also by its attribute `isHist`, as in Fig. A.2. The histogram can afterwards be plotted using the `fields` attribute.

```
# make a request for a simulation
> simId = SECSIM.request({numUEs: 1000, simTimeLimit: 180, numAt-tackers: 300,
  rrcAttack: DCH, useMitCounter: True, consTrans: 3, isMitigating: True,
  blockTime: 60 })
# --- simulation is run ---

# the following steps are same as in a regular simulation request, get the
# simulation object, upon receiving simId parameter
> sim = SECSIM.getSimulation(simId)

# get a list of all possible scalar results for all modules in the network
> listScalars = sim.getScalars()

# get a particular statistic object for UE[0] containing a histogram of
# response times
> stat = getStatisticByModuleAndName(GenericNet.ues[0].appLayer.tcpApps[0],
 responseTime:histogram)
> if stat.isHist:
>    histBins = stat.bins
6.3
```

Further improvements in the remote control of SECSIM could be done in extending the database storage, enabling running of more complex simulations, or building client side data visualisation tools. SECSIM as a simulator, as well as the described remote control, apart from education and research institutions, could also be used by network operators, in particularly security analysts.

A.4 SECSIM evaluation

SECSIM's features are suitable for running the simulations regarding the detection mechanisms. Anyway, in order to understand its performance, a small experiment was run in the scope of this work. The experiment's goal is to benchmark SECSIM's *speed* in terms of processed events per unit time, and its *relative speed* in terms of simulated time units per unit time. The simulator's performance can vary depending on a number of different factors:

- **Hardware**. The performance of the physical machine the simulation is running on. To this regard, number of CPUs and their speed are the main factors.
- **Model size**. Number of components in the model. In a mobile network, this is the number of simulated entities - UEs, RNCs, etc.
- **Model setup**. This factor incorporates the complexity of a network model (UMTS or LTE model) and the complexity of the applications used (web browsing, VoIP, etc.).

In order to calculate the speed and relative speed of SECSIM, a fixed hardware platform is used including an i5 processor with 3.20GHz speed and four cores, 5.7 GB RAM memory and Ubuntu 14.04.3 LTS operating system. The model setup is varied using the following apps: web browsing, SMS, IM, web browsing & RRC attack, SMS & spam. The RRC attack app refers to a signalling attack used in parallel with a legitimate web browsing app. The model size is kept fixed in the experiment, because the simulator's speed does not depend on it.

Fig. A.4 shows SECSIM's speed for different type of simulated models/applications and fixed model size. The speed represented in processed events per second is a measure of the computation intensiveness of different applications. Among the three basic types of applications, web browsing, SMS and IM, the web browsing is most computationally expensive, while SMS is the least expensive to simulate. The combination of web browsing and RRC attack applications results in a slightly lower speed compared to the web browsing

case. This decline is probably due to the fact that the RRC attack application has lower computational complexity, because malicious terminals only occupy the bandwidth and do not transfer any data. The combination of SMS and spam applications results in higher speed than the SMS case. This is due to the higher complexity of the spam application.

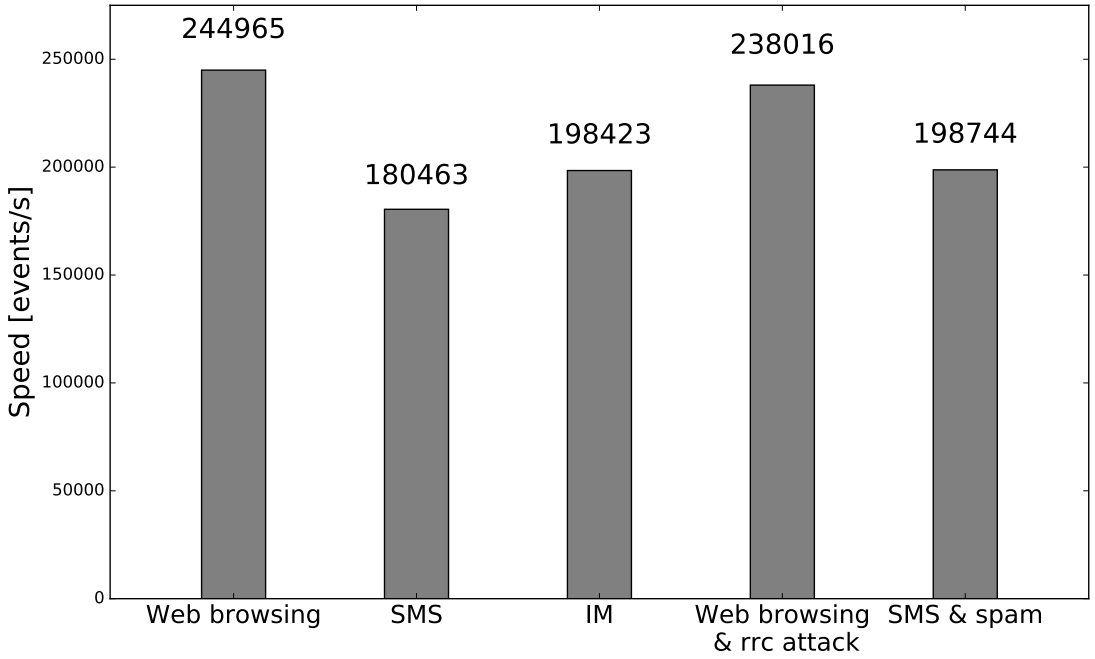


Figure A.4: Speed for different type of application models.

Fig. A.5 shows the relative speed for different application models and model sizes. For the web browsing and web browsing & RRC attacks apps the number of UEs is 100, 500 and 1000, while for the other apps the number of UEs is 1000, 5000 and 10000. The hardware configuration is fixed, as a single processor is used. The two cases that use web browsing application show substantially lower relative speed than the rest of the applications. This indicates that although web browsing performs best in processed events per second (Fig. A.4), it also contains the highest number of events to be processed, which results in low relative speed. The SMS and IM models' relative speed is significantly higher, with IM having the highest score. As expected, the relative speed decreases with the increase of the model size.

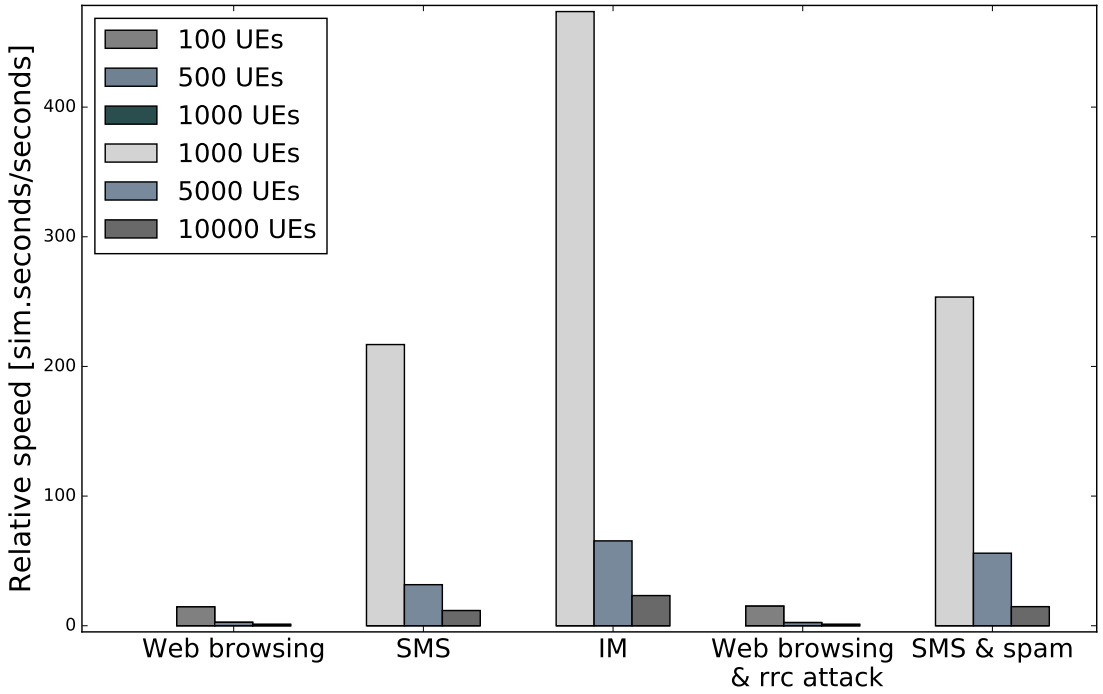


Figure A.5: Relative speed for different types of application models.

The described experiments estimate the simulation running time depending on its configuration, in particular depending on the selected applications running on mobile devices. Anyway, there are many more aspects of the simulator that need to be evaluated, such as the parallel execution of parts of the simulation on different machines, and the processing and memory complexity of different configurations.

List of abbreviations

AES Advanced Encryption Standard.

AI Artificial Intelligence.

ANN Artificial Neural Networks.

API Application Program Interface.

BS Base Station.

CN Core Network.

CPN Cognitive Packet Network.

CS Circuit Switched.

DCH Dedicated Physical Channel.

DDoS Distributed DoS.

DL downlink.

DNN Deep Neural Networks.

DNS Domain Name System.

DoS Denial of Service.

DPI Deep Packet Inspection.

EAP Extensible Authentication Protocol.

EDGE Enhanced Data rates for GSM Evolution.

eNodeB Evolved NodeB.

EWMA Exponential Weighted Moving Average.

FACH Forward Access Channel.

FIFO First In First Out.

GGSN Gateway GPRS Support Node.

GMM GPRS Mobility Management.

GPRS General Packet Radio Systems.

GSMA Global System for Mobile Communications Association.

GTP GPRS Tunneling Protocol.

HSPA High Speed Packet Access.

IM Instant Messaging.

IMSI International Mobile Subscriber Identity.

IoT Internet of Things.

IP Internet Protocol.

IPsec Internet Protocol Security.

ISN Intelligent Systems and Networks.

JSON JavaScript Object Notation.

LTE Long Term Evolution.

M2M Machine to Machine.

MAC Media Access Control Layer.

ML Machine Learning.

MM Mobility Management.

MNO Mobile Network Operator.

NAT Network Address Translation.

NBAP NodeB Application Part.

NN Neural Networks.

PHY Physical Layer.

PS Packet Switched.

PSTN Public Switched Telephone Networks.

QoS Quality of Service.

RAN Radio Access Network.

RANAP Radio Access Network Application Part.

RB resource block.

RB Radio Bearer.

RNC Radio Network Controller.

RNG Random Number Generator.

RNN Random Neural Network.

ROC Receiver Operating Curve.

RRC Radio Resource Control.

RRM Radio Resource Management.

SECSIM Mobile Networks Security Simulator.

SFTP Secure File Transfer Protocol.

SGSN Serving GPRS Support Node.

SGW Serving Gateway.

SIP Session Initiation Protocol.

SM Session Management.

SMS Short Message Service.

SOAP Simple Object Access Protocol.

SS Signaling Server.

SSDM Signaling Storm Detector and Mitigator.

SSH Secure Shell.

TCP Transmission Control Protocol.

TLS Transport Layer Security.

TMSI Temporary Mobile Subscriber Identity.

UDP User Datagram Protocol.

UE User Equipment.

UL uplink.

UMTS Universal Mobile Telecommunications System.

VoIP Voice over IP.

WAPI WOMBAT API.

Bibliography

[1] A. Barbuzzi, F. Ricciato, and G. Boggia, "Discovering Parameter Setting in 3G Networks via Active Measurements," *IEEE Communications Letters*, vol. 12, pp. 730–732, October 2008.

[2] Z. Qian, Z. Wang, Q. Xu, Z. M. Mao, M. Zhang, and Y.-M. Wang, "You can run, but you can't hide: Exposing network location for targeted dos attacks in cellular networks," in *NDSS*, 2012.

[3] O. H. Abdelrahman and E. Gelenbe, "Signalling storms in 3G mobile networks," in *IEEE International Conference on Communications (ICC'14), Communication and Information Systems Security Symposium*, (Sydney, Australia), pp. 1023 –1028, June 2014.

[4] G. Gorbil, O. H. Abdelrahman, and E. Gelenbe, "Storms in mobile networks," in *Proceedings of the 9th ACM Symposium on QoS and Security for Wireless and Mobile Networks (Q2SWinet'14)*, pp. 119–126, 2014.

[5] M. Pavloski and E. Gelenbe, "Signaling attacks in mobile telephony," in *Proceedings of the 11th International Conference on Security and Cryptography (SECRYPT'14)*, pp. 206–212, August 2014.

[6] "Cisco visual networking index: Global mobile data traffic forecast update, 20152020," tech. rep., Cisco, February 3, 2016.

[7] G. Brown, "The Evolution of the Signaling Challenge in 3G & 4G Networks," June, 2012. Cisco white paper.

[8] R. Kalden, I. Meirick, and M. Meyer, "Wireless Internet access based on GPRS," *Personal Communications, IEEE*, vol. 7, no. 2, pp. 8–18, 2002.

[9] "Universal Mobile Telecommunications System (UMTS); Technical Specifications and Technical Reports for a UTRAN-based 3GPP system," TS 21.101, 3rd Generation Partnership Project (3GPP), 2009.

[10] "Requirements for Evolved UTRA (E-UTRA) and Evolved UTRAN (E-UTRAN)," TS 25.913, 3rd Generation Partnership Project (3GPP), 2010.

[11] C. Gabriel, "VoIP signaling crashed NTT DoCoMo; asks Google to help," *Rethink Wireless*, January 2012.

[12] C. Gabriel, "O2 and France Telecom suffer severe outages," *Rethink Wireless*, July 2012.

[13] D.Sahota, "O2 outage causes concern ahead of olympics," *Telecoms.com*, July 2012.

[14] P. Donegan, "Mobile Network Outages & Service Degradations : A Heavy Reading Survey Analysis," October 2013.

[15] Cisco, "Cisco visual networking index: Global mobile data traffic forecast update, 2014–2019." White Paper, 2015.

[16] Y. Choi, C. hyun Yoon, Y.-S. Kim, S. W. Heo, and J. A. Silvester, "The impact of application signaling traffic on public land mobile networks," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 166–172, 2014.

[17] Kaspersky Lab and INTERPOL, "Mobile Cyber Threats." Joint Report, October 2014.

[18] "Mobile malware report," tech. rep., GData, March 2015.

[19] A. Gupta, T. Verma, S. Bali, and S. Kaul, "Detecting ms initiated signaling ddos attacks in 3g/4g wireless networks," *2013 Fifth International Conference on Communication Systems and Networks (COMSNETS)*, pp. 1–60, January 2013.

[20] F. Francois, O. H. Abdelrahman, and E. Gelenbe, "Impact of signaling storms on energy consumption and latency of LTE user equipment," in *Proceedings of the 7th IEEE International Symposium on Cyberspace safety and security (CSS15)*, (New York), 2015.

[21] C. P. Pfleeger and S. L. Pfleeger, *Security in Computing (4th Edition)*. Upper Saddle River, NJ, USA: Prentice Hall PTR, October 2006.

[22] E. Gelenbe and F.-J. Wu, "Future research on cyber-physical emergency management systems," *Future Internet*, vol. 5, no. 3, pp. 336–354, 2013.

[23] A. Kokuti and E. Gelenbe, "Directional navigation improves opportunistic communication for emergencies," *Sensors*, vol. 14, no. 8, pp. 15387–15399, 2014.

[24] E. Gelenbe, M. Gellman, and G. Loukas, "Defending networks against denial of service attacks," in *Proceedings of the Conference on Optics/Photonics in Security and Defence (SPIE), Unmanned/Unattended Sensors and Sensor Networks* (E. Carapezza, ed.), vol. 5611, (London, UK), pp. 233–243, October 2004.

[25] R. Unuchek and V. Chebyshev, "Mobile malware evolution 2015," tech. rep., Kaspersky Lab., February 2015.

[26] K. Kotapati, P. Liu, Y. Sun, and T. F. LaPorta, *Intelligence and Security Informatics: IEEE International Conference on Intelligence and Security Informatics, ISI 2005, Atlanta, GA, USA, May 19-20, 2005. Proceedings*, ch. A Taxonomy of Cyber Attacks on 3G Networks, pp. 631–633. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[27] L. Spitzner, "Honeypots: catching the insider threat," in *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*, pp. 170–179, December 2003.

[28] V. Igure and R. Williams, "Taxonomies of attacks and vulnerabilities in computer systems," *Communications Surveys Tutorials, IEEE*, vol. 10, pp. 6–19, First 2008.

[29] S. Mavoungou, G. Kaddoum, M. Taha, and G. Matar, "Survey on threats and attacks on mobile networks," *IEEE Access*, vol. 4, pp. 4543–4572, 2016.

[30] U. Meyer and S. Wetzel, "On the impact of gsm encryption and man-in-the-middle attacks on the security of interoperating gsm/umts networks," in *Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium on*, vol. 4, pp. 2876–2883 Vol.4, September 2004.

[31] E. Gelenbe, "Steps toward self-aware networks," *Commun. ACM*, vol. 52, no. 7, pp. 66–75, 2009.

[32] E. Gelenbe, "Self-aware networks," in *Self-Adaptive and Self-Organizing Systems (SASO), 2011 Fifth IEEE International Conference on*, pp. 227–234, IEEE, 2011.

[33] G. Loukas, G. Oke, E. Gelenbe, *et al.*, "Defending against denial of service in a self-aware network: A practical approach," in *NATO Symposium on Information Assurance for Emerging and Future Military Systems. Ljubljana, Slovenia*, 2008.

[34] E. Gelenbe and G. Loukas, "A self-aware approach to denial of service defence," *Comput. Netw.*, vol. 51, no. 5, pp. 1299–1314, 2007.

[35] E. Gelenbe, "Search in unknown random environments," *Physical Review E*, vol. 82, no. 6, p. 061112, 2010.

[36] E. Gelenbe and O. Abdelrahman, "Search in the universe of big networks and data," *IEEE Network*, vol. 28, no. 4, pp. 20–25, 2014.

[37] O. H. Abdelrahman and E. Gelenbe, "Search in big networks and big data," in *Analytic Methods in Interdisciplinary Applications*, pp. 1–15, Springer International Publishing, 2015.

[38] E. Gelenbe, M. Gellman, and G. Loukas, "Defending networks against denial-of-service attacks," 2004.

[39] F. Ricciato, A. Coluccia, and A. DAlconzo, "A review of DoS attack models for 3G cellular networks from a system-design perspective," *Computer Communications*, vol. 33, no. 5, pp. 551 – 558, 2010.

[40] "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access," TS 25.913, 3rd Generation Partnership Project (3GPP), December 2009.

[41] R. Chandramouli and S. Rose, "Challenges in securing the domain name system," *IEEE Security Privacy*, vol. 4, pp. 84–87, January 2006.

[42] Yaniv Balmas, "Mobile network security, availability risks in mobile networks," 2013. ERT Lab.

[43] J. D. et al., "D2.2 honeydroid: Virtualized mobile honeypot for android," public deliverable, NEMESYS Project, November 2014.

[44] F. D. et al., "Lte security," 2011.

[45] D. Maslennikov, "Mobile malware evolution: Part 6," tech. rep., Kaspersky Lab., February 2013.

[46] F-Secure, "Threat report 2015," 2016.

[47] M. Labs, "Mcafee labs threats report," March 2016.

[48] Lookout, "2014 mobile threat report," 2014.

[49] L. Delosières and A. Sánchez, *Information Sciences and Systems 2014: Proceedings of the 29th International Symposium on Computer and Information Sciences*, ch. DroidCollector: A Honeyclient for Collecting and Classifying Android Applications, pp. 175–183. Cham: Springer International Publishing, 2014.

[50] 3rd Generation Partnership Project (3GPP), "Radio Resource Control (RRC); Protocol specification," TS 25.331.

[51] S. A. P. GSMA, "Smart apps for smarter phones," April 2012.

[52] Stoke Inc., "Charting the signaling storms," 2013. White paper.

[53] Qualcomm, "Managing Background Data Traffic in Mobile Devices," tech. rep., Qualcomm, January, 2012. White paper.

[54] H. T. Co., "Smartphone solutions," tech. rep., Huawei Technologies Co., July 2012.

[55] Arbor Networks Inc., "Arbor Networks Unveils Peakflow Mobile Network Analysis," February 2014.

[56] E. Gelenbe, G. Gorbil, D. Tzovaras, S. Liebergeld, D. Garcia, M. Baltatu, and G. Lyberopoulos, "Security for smart mobile networks: The NEMESYS approach," in *Proceedings of the 2013 IEEE Global High Tech Congress on Electronics (GHTCE'13)*, pp. 63–69, 2013.

[57] O. H. Abdelrahman, E. Gelenbe, G. Gorbil, and B. Oklander, "Mobile network anomaly detection and mitigation: The NEMESYS approach," in *Information Sciences and Systems 2013 - Proceedings of the 28th International Symposium on Computer and Information Sciences (ISCIS'13)* (E. Gelenbe and R. Lent, eds.), vol. 264 of *Lecture Notes in Electrical Engineering*, pp. 429–438, Springer, 2013.

[58] G. Kambourakis, C. Kolias, S. Gritzalis, and J. H. Park, "Dos attacks exploiting signaling in umts and ims," *Comput. Commun.*, vol. 34, no. 3, pp. 226–235, 2011.

[59] F. Qian, Z. Wang, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "Characterizing radio resource allocation for 3g networks," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, IMC '10, (New York, NY, USA), pp. 137–150, ACM, 2010.

[60] F. Francois, O. H. Abdelrahman, and E. Gelenbe, "Feasibility of signaling storms in 3G/UMTS operational networks," in *Internet of Things. IoT Infrastructures: Second International Summit, IoT 360 2015, Rome, Italy, October 27-29, 2015. Revised Selected Papers, Part I*, pp. 187–198, Springer International Publishing, 2016.

[61] G. Gorbil, O. H. Abdelrahman, M. Pavloski, and E. Gelenbe, "Modeling and analysis of RRC-based signaling storms in 3G networks," *IEEE Transactions on Emerging Topics in Computing, Special Issue on Emerging Topics in Cyber Security*, vol. PP, no. 99, pp. 1–14, 2015.

[62] F. Francois, O. H. Abdelrahman, and E. Gelenbe, "Towards assessment of energy consumption and latency of lte ues during signaling storms," in *Information Sciences and Systems 2015*, pp. 45–55, Springer, 2016.

[63] E. Gelenbe and O. H. Abdelrahman, "Countering mobile signaling storms with counters," *Proc. Intl Conf. on Cyber Physical Systems, IoT and Sensors Networks (Cyclone), Rome, Italy*, 2015.

[64] M. Pavloski, G. Görbil, and E. Gelenbe, "Counter based detection and mitigation of signalling attacks," in *SECRYPT 2015 - Proceedings of the 12th International Conference on Security and Cryptography, Colmar, Alsace, France, 20-22 July, 2015.* (M. S. Obaidat, P. Lorenz, and P. Samarati, eds.), pp. 413–418, SciTePress, 2015.

[65] E. Gelenbe, O. H. Abdelrahman, and G. Gorbil, "Detection and mitigation of signaling storms in mobile networks," in *Computing, Networking and Communications (ICNC), 2016 International Conference on*, pp. 1–5, IEEE, 2016.

[66] P. P. C. Lee, T. Bu, and T. Woo, "On the detection of signaling DoS attacks on 3G wireless networks," in *Proceedings - IEEE INFOCOM*, pp. 1289–1297, 2007.

[67] H. Wang, D. Zhang, and K. Shin, "Change-point monitoring for the detection of dos attacks," *Dependable and Secure Computing, IEEE Transactions on Dependable and Secure Computing (Volume:1 , Issue: 4)*, vol. 1, pp. 193–208, October 2004.

[68] Z. Wu, X. Zhou, and F. Yang, "Defending against DoS attacks on 3G cellular networks via randomization method," in *Educational and Information Technology (ICEIT), 2010 International Conference on*, vol. 1, pp. V1–504–V1–508, 2010.

[69] C. Mulliner, S. Liebergeld, M. Lange, and J. P. Seifert, "Taming Mr Hayes: Mitigating signaling based attacks on smartphones," in *Proceedings of the International Conference on Dependable Systems and Networks*, 2012.

[70] M. Pavloski, G. Görbil, and E. Gelenbe, *Information Sciences and Systems 2015: 30th International Symposium on Computer and Information Sciences (ISCIS 2015)*, ch. Bandwidth Usage—Based Detection of Signaling Attacks, pp. 105–114. Cham: Springer International Publishing, 2016.

[71] O. H. Abdelrahman and E. Gelenbe, "A data plane approach for detecting control plane anomalies in mobile networks," in *Internet of Things. IoT Infrastructures: Second International Summit, IoT 360 2015, Rome, Italy, October 27-29, 2015. Revised Selected Papers, Part I*, pp. 210–221, Springer International Publishing, 2016.

[72] O. H. Abdelrahman, "Detecting network-unfriendly mobiles with the random neural network," *Probability in the Engineering and Informational Sciences*, vol. 30, no. 3, pp. 514–531, 2016.

[73] G. Gorbil, O. H. Abdelrahman, and E. Gelenbe, "Storms in mobile networks," in *Proceedings of the 10th ACM symposium on QoS and security for wireless and mobile networks*, pp. 119–126, ACM, 2014.

[74] G. Oke, G. Loukas, and E. Gelenbe, "Detecting denial of service attacks with bayesian classifiers and the random neural network," in *Fuzzy Systems Conference, 2007. FUZZ-IEEE 2007. IEEE International*, pp. 1–6, IEEE, 2007.

[75] E. Gelenbe and I. Mitrani, "Analysis and synthesis of computer systems," vol. Vol. 4, 2010.

[76] E. Gelenbe, "Probabilistic models of computer systems," *Acta Inf.*, vol. 12, pp. 285–303, 1979.

[77] H. S. Dhillon, H. C. Huang, H. Viswanathan, and R. A. Valenzuela, "Throughput optimal communication strategy for wireless random access channel," in *Global Communications Conference (GLOBECOM), 2013 IEEE*, pp. 2026–2031, IEEE, 2013.

[78] N. Abramson, "The aloha system: Another alternative for computer communications," in *Proceedings of the November 17-19, 1970, Fall Joint Computer Conference*, AFIPS '70 (Fall), (New York, NY, USA), pp. 281–285, ACM, 1970.

[79] I. Gitman, "On the capacity of slotted aloha networks and some design problems," *IEEE Transactions on Communications*, vol. 23, no. 3, pp. 305–317, 1975.

[80] J. R. Jackson, "Jobshop-like queueing systems," *Management Science*, vol. 10, no. 1, pp. 131–142, 1963.

[81] G. F. N. William J. Gordon, "Closed queuing systems with exponential servers," *Operations Research*, vol. 15, no. 2, pp. 254–265, 1967.

[82] F. Baskett, K. M. Chandy, R. R. Muntz, and F. G. Palacios, "Open, closed, and mixed networks of queues with different classes of customers," *J. ACM*, vol. 22, no. 2, pp. 248–260, 1975.

[83] E. Gelenbe, "Product-form queueing networks with negative and positive customers," *Journal of Applied Probability*, vol. 28, no. 3, pp. 656–663, 1991.

[84] Erol Gelenbe, "G-networks with triggered customer movement," *Journal of Applied Probability*, vol. 30, no. 3, pp. 742–748, 1993.

[85] P. G. Harrison, "Turning back time in markovian process algebra," *Theoretical Computer Science*, vol. 290, no. 3, pp. 1947 – 1986, 2003.

[86] P. G. Harrison, "Reversed processes, product forms and a non-product form," *Linear Algebra and its Applications*, vol. 386, pp. 359 – 381, 2004. Special Issue on the Conference on the Numerical Solution of Markov Chains 2003.

[87] P. G. Harrison, "Product-forms and functional rates," *Performance Evaluation*, vol. 66, no. 11, pp. 660 – 663, 2009.

[88] P. Harrison, "Compositional reversed markov processes, with applications to g-networks," *Performance Evaluation*, vol. 57, no. 3, pp. 379 – 408, 2004.

[89] S. Balsamo, P. G. Harrison, and A. Marin, "A unifying approach to product-forms in networks with finite capacity constraints," *SIGMETRICS Perform. Eval. Rev.*, vol. 38, no. 1, pp. 25–36, 2010.

[90] S. Balsamo, R. O. Onvural, and V. D. N. Persone, *Analysis of Queueing Networks with Blocking*. Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[91] E. Gelenbe, G. Görbil, D. Tzovaras, S. Liebergeld, D. Garcia, M. Baltatu, and G. Lyberopoulos, "Nemesys: Enhanced network security for seamless service provisioning in the smart mobile ecosystem," in *Information Sciences and Systems 2013*, pp. 369–378, Springer International Publishing, 2013.

[92] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," in *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*, Simutools '08, (ICST, Brussels, Belgium, Belgium), pp. 60:1–60:10, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[93] S. Ramachandran, "Web metrics: Size and number of resources," May 2010.

[94] I. Kalamaras, A. Drosou, and D. Tzovaras, "A multi-objective clustering approach for the detection of abnormal behaviors in mobile networks," in *2015 IEEE International Conference on Communication Workshop (ICCW)*, pp. 1491–1496, June 2015.

[95] J. Korhonen, "Introduction to 3g mobile communications," 2003.

[96] I. Poole, *Cellular Communications Explained: From Basics to 3G*. Electronics & Electrical, Newnes, 2006.

[97] H. Holma and A. Toskala, *WCDMA for UMTS: HSPA Evolution and LTE*. John Wiley Sons, Ltd, 2010.

[98] I. Chantaksinopas, W. Lee, A. Prayote, and P. Oothongsap, "Delay-sensitive applications in vanet and seamless connectivity: The limitation of umts network," *IJCSNS International Journal of Computer Science and Network Security*, October 2012.

[99] Nokia, "What is going on in Mobile Broadband Networks? Smartphone Traffic Analysis and Solutions," tech. rep., September 2014. White paper.

[100] A. K. Erlang, "The theory of probabilities and telephone conversations," *Nyt Tidsskrift for Matematik*, vol. 20, no. B, pp. 33–39, 1909.

[101] J. W. Cohen, "The multiple phase service network with generalized processor sharing," *Acta Informatica*, vol. 12, no. 3, pp. 245–284, 1979.

[102] S.-K. Cheung, *Processor-sharing queues and resource sharing in wireless LANs*. PhD thesis, University of Twente, 2007.

[103] Real Wireless Ltd., "Report for Ofcom: 4G capacity gains," 2011.

[104] 3rd Generation Partnership Project (3GPP), "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation (Rel 8)," TS 36.211, ETSI, 2010.

[105] Huawei, "UMTS RAN10.0 Dimensioning Rules," tech. rep., Huawei Technologies Co., Ltd., 2008.

[106] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, S. Venkataraman, and J. Wang, "A first look at cellular network performance during crowded events," *SIGMETRICS Perform. Eval. Rev.*, vol. 41, no. 1, pp. 17–28, 2013.

[107] 3GPP, "Study on machine-type communications (mtc) and other mobile data applications communications enhancements (release 12)," 2013. 3GPP TR 23.887.

[108] 3GPP, "TR 37.869: Study on enhancements to machine-type communications (MTC) and other mobile data applications; radio access network (RAN) aspects (Release 12)." http://www.3gpp.org/DynaReport/37869.htm, 2013. Technical Specification Group Radio Access Network.

[109] T. Taleb and A. Kunz, "Machine type communications in 3GPP networks: potential, challenges, and solutions," *IEEE Communications Magazine*, vol. 50, no. 3, pp. 178–184, 2012.

[110] A. Kunz, L. Kim, H. Kim, and S. Husain, "Machine type communications in 3GPP: From release 10 to release 12," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, pp. 1747–1752, 2012.

[111] R. Ratasuk, A. Prasad, Z. Li, A. Ghosh, and M. A. Uusitalo, "Recent advancements in M2M communications in 4G networks and evolution towards 5G," in *Proc. 18th IEEE International Conference Intelligence in Next Generation Networks (ICIN)*, (Paris, France), pp. 52–57, 2015.

[112] K. Gurney, *An introduction to neural networks*. UCL Press, 1 ed., 1997.

[113] E. Gelenbe, "Random neural networks with negative and positive signals and product form solution," *Neural Comput.*, vol. 1, no. 4, pp. 502–510, 1989.

[114] E. Gelenbe, "Learning in the recurrent random neural network," *Neural Comput.*, vol. 5, no. 1, pp. 154–164, 1993.

[115] S. Timotheou, "The Random Neural Network: A Survey," *The Computer Journal*, 2009.

[116] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. Software available from tensorflow.org.

[117] J. Schmidhuber, "Deep learning in neural networks: An overview," *CoRR*, vol. abs/1404.7828, 2014.

[118] E. Kirda, "Wombat Deliverable D12/D5.1 Root Causes Analysis," public deliverable, FP7-ICT-216026-WOMBAT, 2007-2013.

[119] G. van Rossum and F. Drake, "Python Reference Manual," 2001.

[120] "MongoDB." Open-source document database.