AUTOMATED TROUBLESHOOTING FOR RTWP IN 3G/4G RAN NODES

by

Hisham Mohammed

MHMHIS002

Thesis submitted in partial fulfillment of the academic requirements for the Degree of
Master of Engineering in Telecommunications
in the Faculty of Engineering and the Built Environment
University of Cape Town

Supervisor: Dr S. Winberg

December 2017

As the candidate's supervisor, I have approved this dissertation for submission.

Name: Dr Simon Winberg

Signed: 

Date: 1 December 2017

DECLARATION

I declare that the work presented in this dissertation is original and my own. Ideas and material generated from other researchers are explicitly stated with appropriate references.

This work is submitted for the Master of Engineering specializing in Telecommunication at the University of Cape Town. It has not been submitted to any other university for any other degree or examination.

Name: Hisham Mohammed

Signature: Signed by candidate

Date: 1 December 2017

ABSTRACT

Nowadays, Mobile Network Operators are confronted with many challenges to operate and maintain their network. Subscribers expect stable and perpetual services. Repeated interruptions of services will result in the dissatisfaction of users and may lead to losing the end user. One of the major issues facing a Radio Access Network (RAN) mobile operator is coping with the uplink interference in their RAN, such as the Receive Total Wideband Power (RTWP) in the Universal Mobile Telecommunications System (UMTS) band. A frequently occurring issue in such networks is the RTWP alarm. This alarm is reported in the Network Operation Centre (NOC) and contribute to poor quality in the network . Such an alarm may occur daily, thus impacting the network's Key Point Indicator (KPI). The mobile network operator always tries to resolve the issue of RTWP quickly by means of several processes and strategies to diagnose and troubleshoot this issue, all within a target 'Service Level Agreement' (SLA). There are many different causes that can lead to an RTWP alarm in a mobile 3G RAN. In addition, each of these cases has different diagnoses and troubleshooting methods. The main idea of this project is to design a Graphical User Interface (GUI) tool to help the Front Office (FO) or Back Office (BO) engineer in mobile network operator to check and troubleshoot the RTWP issue in the network in a timely manner. The tool is designed to check the configuration of the radio, based on the Huawei NodeB 3900 and statistical performance counters, and to provide the correct decision for the engineer to improve the efficiency and minimize the time taken to troubleshoot the RTWP alarm in the network. It is very useful to design such a tool for interacting with the Huawei NodeB 3900. The GUI tool is thus basically designed to support the engineers in Oman Telecommunication Company's NOC while dealing with the RTWP alarm in the Huawei NodeB 3900. The major finding of this study is the design of the GUI tool to minimize the time taken to resolve the RTWP issue in the Huawei NodeB 3900 both in a single site and in multiple sites, to conduct consistency checks for the software parameters, and finally to identify the root cause of the RTWP alarm. The GUI tool shows an operation log, which can be used by the administrator for maintenance records, and it also contains a help guide that gives the user more information about the functionality of each button.

ACKNOWLEDGEMENTS

I thank my supervisor, Dr Simon Winberg, for his encouragement, constructive criticism, direction and support in completing this research work. The experience of working with Dr Winberg has always been delightful and enlightening. I feel privileged to have worked with him.

I also thank my caring and supportive family, especially my wife, Hiba Eisa, and my daughter, Mayar, for always uplifting my spirit and cheering me on to work as hard as I could.

Finally, I thank the Almighty God for giving me strength and courage to never fear my weaknesses, but learn from them to grow personally and professionally.

TABLE OF CONTENTS

LIST OF ACRONYMS

3GPP	3rd Generation Partnership Project
ADC	Analog to Digital Conversion
AMR	Adoptive Multi Rate
BBU	Base Band Unit
BO	Back Office
BTS	Base Transvers Station
CAPEX	Capital Expenditure
CE	Channel Element
CEU	Cell Edge Users
CN	Core Network
DAC	Digital Conversion to Analog
DAGC	Digital Automatic Gain Control
DCH	Dedicated Channel
DL	Downlink
FE	Fast Ethernet
FM	Field Maintenance
FO	Front Office
G	Gain
GIS	Geographical Information System
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSGSN	Gateway GPRS Support Node
GSM	Global System for Mobile
GUI	Graphical User Interface
HLR	Home Location Registration
ICT	Information Communication Technology
IF	Intermediate Frequency
KPI	Key Point Indicator
LAC	Location Area Code
LMT	Local Maintenance Terminal
LNA	Low Noise Amplifier
LTE	Long-Term Evolution
MATLAB	Matrix Laboratory
MGW	Media Gateway
MSC	Mobile Switching Centre
NF	Noise Factor
NOC	Network Operation Centre
OAM	Operation and Maintenance
OM	Operation and Maintenance
OMC	Operation Maintenance Centre
OPEX	Operational Expenditure
OSS	Operation Switching Subsystem
PA	Power Amplifier
PIM	Passive Intermodulation

PN	Noise Power
PoC	Proof of Concept
PS	Packet Switch
QoS	Quality of Service
R&D	Research and Development (R&D)
RAN	Radio Access Network
RF	Radio Frequency
RNC	Radio Network Controller
RNO	Radio Network Optimization
RNP&O	Radio Network Planning & Optimization
RNS	Radio Network Subsystems
RoT	Rise over Thermal
RRU	Radio Remote Unite
RSCP	Receive Signal Code Power
RTWP	Receive Total Wideband Power
	Receiver
SAC	Service Area Code
SGSN	Serving GPRS Support Node
SLA	Service Level Agreement
SON	Self-Organizing Networks
SSV	Single Site Verification
TCP	Transmit Code Power
TMA	Tower Mountain Amplifier
TRA	Telecommunication Regulation Authority
TRX	Transceiver
TX	Transmitter
UE	User Equipment
UL	Uplink
UML	Unified Model Language
UTRP	Universal Transmission Possessing Unit
VIP	Very Important
VLR	Visitor Location Register
WCDMA	Wideband Code Deviation Multiple Access
WMPT	WCDM Main Process and Timing
WRFU	WCDM Radio Frequency Unit
WBBP	WCDMA Base Band Processing Unit
PIM	Passive intermodulation interference
RET	Remote Electrical Tilt
UTRAN	UMTS Terrestrial Radio Access Network

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

This dissertation describes the design of a Graphical User Interface (GUI) tool to support fault finding and maintenance of cellular transceiver base stations that use the Huawei NodeB radio access module. The specific details of the problem to be addressed are explained in Section 1.2, and the specific objectives of this dissertation are explained in Section 1.3. This chapter proceeds with a brief background outlining radio access technology and common maintenance concerns.

1.1 Background

The evolution in radio access technology in recent years was the result of boosting demand for mobile internet and wireless multimedia applications. Mobile services play a very effective role in the telecommunications industry. The radio access network technologies, standardized by the 3rd Generation Partnership Project (3GPP), are Long-Term Evolution (LTE) and (UMTS). Both provide mobile voice, video and data services with different data rates to mobile subscribers [34]. Most mobile operators currently deploy 3G and 4G technology in their networks and must invest a great deal of effort and money to operate, maintain and optimize these networks to ensure satisfactory service quality for each user [1]. Mobile telecommunication operators rely on a Network Operation Centre (NOC) to provide network precautionary, network corrective, routine maintenance, service provision and site information management services for operators. Some of the activities generally performed in the NOC include:

- Site preventative or precautionary maintenance: Routine preventative maintenance, Special preventative maintenance.
- Site corrective maintenance: Fault Monitor, Level 1, Level 2 support, Work order management and trouble ticket.
- Scheduled or routine maintenance: Network change and configuration management, Equipment change and configuration management.
- Site information management: Site rent information management, Site access information management, Site assets information management.

A centralized NOC, therefore, helps the mobile network operators to manage their networks in an efficient way. With the emergence and the growth of modern technologies, operators face increasing network complexity and increasing financial pressure. Some of the solutions provided by the vendors in the telecommunication industry to improve the operating efficiency and reduce operating expenses (OPEX) include automated tools, centralized offshoring solutions, and outsourcing, to help operators manage the networks efficiently [35].

1.2 Problem Statement

Mobile telecommunication operators are restricted by many regulations from the Telecommunication Regulatory Authority (TRA), such as a (SLA), for clearing minor alarms, major alarms, and critical alarms. One of the major difficulties in mobile network operation are the alarms that occur daily and that require troubleshooting by different departments. Some of the common alarms occur on an almost daily basis in networks that utilises the 3G/4G (RAN). Sometimes the target SLA to clear RTWP alarm exceed the threshould since it requires collaborative work from different departments, such as field maintenance, radio optimization, and logistics. The following are the reasons why RTWP alarm are triggered in the network:

- External interference
- Configuration (Cell, and Tower Mountain Amplifier (TMA) related configuration)
- Antenna feeder Passive intermodulation interference (PIM)
- Antenna feeder connections (cross-connections)
- Heavy traffic in busy sites

Operators and vendors define many processes for coping with RTWP issues on their network, since it has an enormous impact on network performance. Figure 1.1 is an example of the process defined by Huawei to clear the RTWPissue.

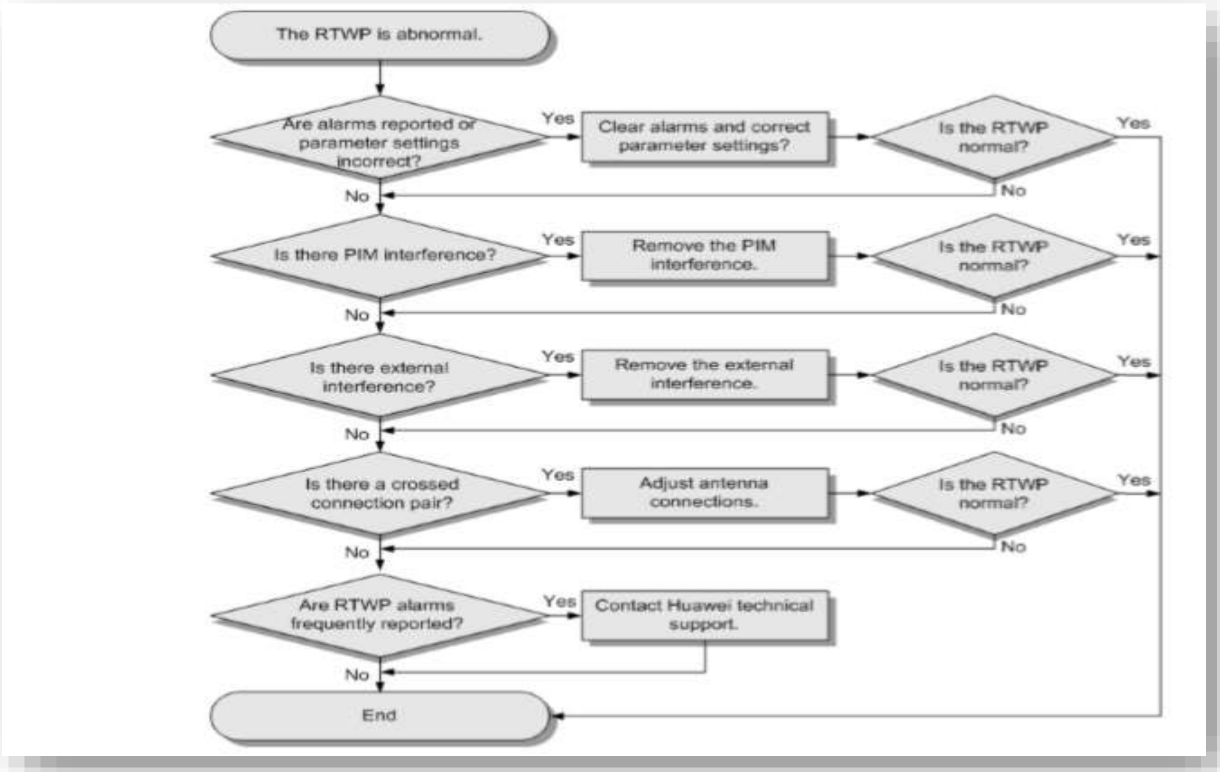


Figure 1.1 RTWP alarm process flow chart [2]

Below are some of the impacts on the RAN's Key Point Indicator (KPI) when RTWP issues appear in the network:

- Uplink coverage is limited , and the users cannot access the network at the edge of the cell or they are dropped from the network.
- Network capacity is restricted, and user experience becomes poor.
- KPI dererioration; for instance ,Figure 1.2 shows that,when Packet Throughput Uplink (UL) or Downlink (DL) increase , it also increase the RTWP.

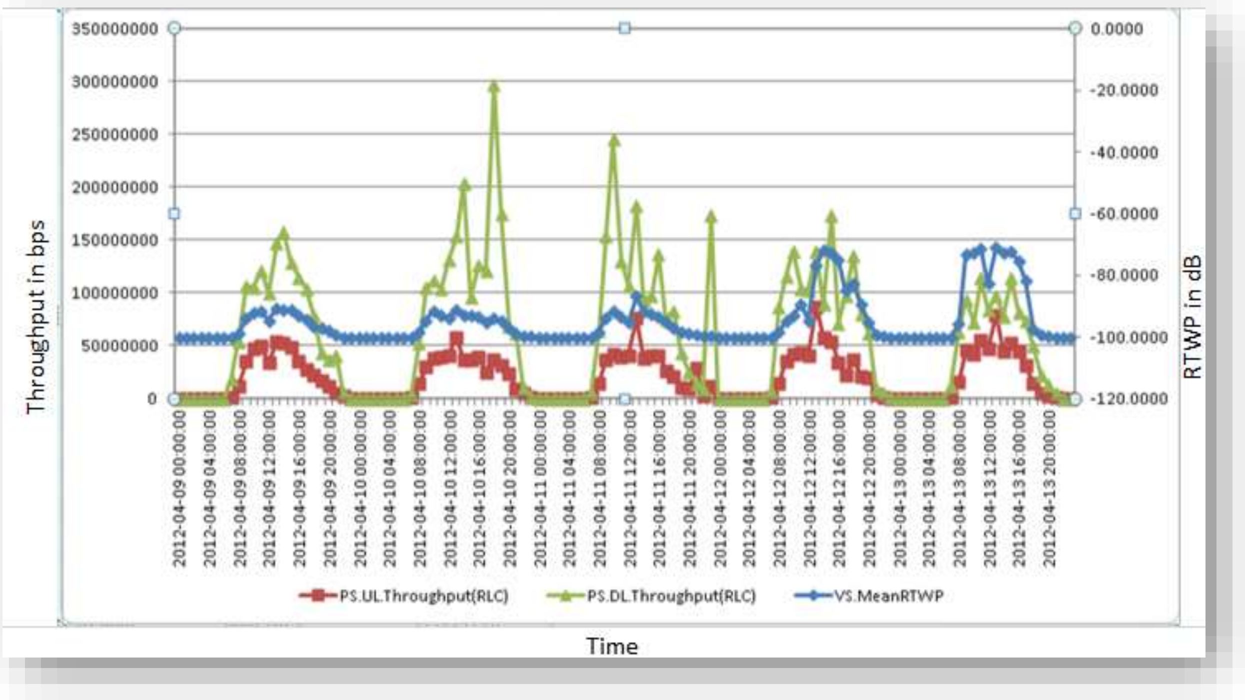


Figure 1.2 Impact of RTWP alarms on KPI's [3]

1.3 Aims And Objectives

The purpose of this work is to design a system with a GUI that reads specific data from the Huawei NodeB, relating to the RTWP issue, by checking the configuration, such as the Receiver (RX) attenuation, TMA and related interference parameters, as well as board configuration, and adjusting the wrong configuration if found. The GUI also checks the user's number that is currently served by the site, to give advice with regard to the troubleshooting status and to send notifications for further analysis. Some of the objectives are listed below:

- The GUI is required to improve the efficacy of the manual technique, i.e. to reduce the delay in communication and reaction between different departments when solving RTWP alarm on the RAN side.
- The GUI is required to assist the various users, such as the Front Office and the Back Office engineers in identifying the root cause of the RTWP alarm and to escalate this directly to the right department within the shortest possible time if the issue is not caused by software configuration , which will improve the end user experience and maintain network quality.

- The researcher is required to develop a proper tool that can be used by the NOC of the mobile network operator that will help to improve the efficacy of troubleshooting RTWP issues, and that can be extended to deal with any daily alarm that has a severe impact on the performance of the 3G/4G radio network.

- The processes needed to carry out requirement changes to the GUI Tool for either supporting an alternate base station or making changes to the RTWP troubleshooting steps must be understood.

1.4 Scope

This study focuses on developing a GUI system that will help the operation engineer to solve RTWP issues in the RAN. The GUI system will allow users to interact with the Huawei 3900 NodeB.

The configuration file will be checked and the configuration will be adjusted, if some of the parameters are not configured according to the planned parameters or the baseline configuration. Moreover, technical advice will be provided to the technical support engineer who is dealing with this issue. In addition, the GUI system needs to be used for a single site or for multiple sites in the network. RTWP alarms can affect a cluster of sites, which is a clear symptom of external interference . Furthermore, the GUI will need to save the operation log so that it can be used for backtracking or investigation, when the end user has misused or abused it. The proposed solutions focus mainly on the following:

- Checking the number of users on the single site or the multiple sites that report the RTWP alarms.
- Checking the configuration parameters related to the RTWP alarms in the NodeB.
- Adjusting the misconfigured parameter.
- Resetting the board in Huawei NodeB.
- Displaying a notification about the current troubleshooting status.
- Sending an email with regard to external interference if found.

1.5 Limitations

This thesis discusses the issue of RTWP in RAN and to design GUI tools that will help the operation engineer to solve the RTWP alarms that can occur in the Huawei 3900 NodeB. The research efforts are concentrated on the GUI tool and how it is to deal with RTWP alarms, in the case of a single-site issue or a multiple-site issue. With regard to the single-site issue, this author shows the capability of the tool for checking the user's number and some related configurations to the cell level, when these are related to the RTWP issue. In addition, the GUI tool will send a notification, if interference is suspected after all the hardware and software solutions have been checked. For a multiple-site solution, this author will check the number of users and the interference status in the list of sites, and perform several commands that could help to resolve the issue. Otherwise, it will send a notification to the Radio Frequency (RF) department to investigate the issue and identify the possible sources of the interference. The dissertation mainly focuses on the operation and maintenance aspects.

1.6 Dissertation Outline

Chapter 1 contains an introduction to the effect of the network's KPI when RTWP alarms are triggered on the network; it also explains how the proposed GUI tool will help the operation team to solve this issue in the live network and to identify and analyse the root cause. It also introduces the impact of RTWP alarms on the 3G RAN KPI's, and provides a general overview of the NOC process for resolving RTWP alarms. The goals and purposes of this study are also introduced in this chapter, as are the scope of the dissertation and the limitations of the research.

Chapter 2 discusses in detail the theoretical background of RTWP and its impact on the 3G live network, including mathematical calculations and the root causes of RTWP alarms, the role of TMA in the NodeB, the mechanism used by most telecommunication operators to cope with interference, and how NOC interacts with this alarm when it appears on a monitor system. For example, a structure for the NOC in the Oman Telecommunication Company is discussed in this chapter. In addition, the chapter introduces the UMTS and important Nodes that are related to this issue, as well as hardware-related boards in Huawei NodeB 3900; it also explains how Huawei NodeB 3900 are commissioned and integrated into a RAN to deliver 3G service to network subscribers, and the tools that are used in the NOC to troubleshoot transport and radio issues. Furthermore, the chapter discusses the definitions and main classes of Unified Model Language (UML), as well as some advantages of MATLAB over Python; there is also some discussion of various features that are currently used in 4G and proposed for use in the 5G network related to this study.

Chapter 3 presents the methodology that is proposed for the implementation and the framework for designing the GUI tool, starting from understanding the requirements of the end user until the tool reaches the maintenance stage. An open source development process is introduced in this chapter as a research methodology. Two deployment scenarios presented show the compatibility of the GUI tool, which is developed in MATLAB version 2014a; it is tested with MATLAB version 2017a. The life cycle of the developed solution and how it can be adjusted or modified in the future to maintain continuation of the solution developed, is also discussed in Chapter 3. An example is included for adjusting some of the command code in the tool to check various parameters in Ericsson NodeB.

Chapter 4 discusses the system diagram for the GUI tool, presents class diagrams to cover the software design part, and introduces sequence diagrams that explain the interaction between the user and the tools. In addition, activity diagrams and deployment diagrams are also introduced.

Chapter 5 presents the physical and logical structure of Huawei NodeB 3900 and the functionality of each board in NodeB; it explains how the NodeB is commissioned, and discusses the service verification stage after NodeB has been commissioned and integrated with the Radio Network Controller (RNC); lastly, it also shows how the tested site is transferred from the project stage to the operation and maintenance team.

Chapter 6 presents the complete implementation of the GUI tool: it is tested for the first scenario when the RTWP occurs in a single site and when troubleshooting a single-site issue. Moreover, all the possible root causes that can contribute to this issue, additional to the multiple-site scenario, are tested. The presence of interference and the absence of interference are tested, as well as how the tool escalates the issue to the RF team for further troubleshooting.

Chapter 7 contains the results of the analysis of implementing the GUI tool in relation to a single-site issue and presents a comparison between the manual way of clearing a RTWP alarms in a single site and using GUI tool . It introduces the benefits of the solution offered, by looking at the scenario of two sites being affected by interference; it also describes how the tool helps the operation engineer to check some of the performance parameters and execute the commands on the Huawei NodeB 3900 for troubleshooting, as well as looking at how the issue is escalated to the RF team for further analysis. A scenario is presented of two sites suffering from interference and it is shown how the tool helps the operation engineer to increase the power to overcome this issue in two sites simultaneously. A scenario is also discussed where one site is affected by high RTWP: in this regard, it is explained how the tool checks the configuration for the site and how it identified the parameters that were not configured according to the required standard, which was the root cause of the issue; it is also explained how the issue at the site was cleared after the configuration was adjusted. The chapter concludes by presenting a final scenario for a site suffering from high RTWP during peak hours, demonstrating how the tool helps to check the capacity of the site.

Chapter 8 discusses the outcomes of the research and the findings. It discusses the technologies used to achieve the objectives, and the technique for overcoming the problem statement. Future work and recommendations for further enhancement are also presented.

1.7 Chapter Summary

This chapter outlined the background to the GUI on which this dissertation is founded. The chapter looked at some of the challenges faced during operation and maintenance in a telecom mobile operator network. It also defined how this research study plans to resolve these challenges. The aims and objectives, the scope of the dissertation and its limitations were also presented. Lastly, this chapter concluded by giving a brief outline of all the chapters in this study.

CHAPTER 2: LITERATURE REVIEW AND THEORY

2.1 Introduction and Important Terminology

This chapter presents in detail the theory and technologies on which the practical study is based with regard to solving the RTWP issue for the Huawei NodeB 3900 in RAN. Background information is presented on the interference in 3G RAN as well as scenarios where a RTWP increase in the network is found, how it is measured mathematically, the reasons behind such an increase and how it is resolved. In addition, the role of the Tower Mounting Amplifier (TMA) in NodeB and how it works, as well as the benefits of this device in the network and how it contributes to total NodeB gain are also discussed. This chapter also looks at the role and structure of the NOC in the mobile telecommunication company, and at the activities performed by the engineers when the alarm is reported and how they clear RTWP alarms in the network. This is followed by an overview of the UMTS network, in which the functions of each node are introduced, since the design solution helps the engineer working in a NOC to cope with RTWP alarms in UMTS networks. The solution is designed to help the telecommunication operators and vendors to improve their efficiency in resolving RTWP errors when these occur in the network. A brief introduction follows to the Unified Model Language (UML) and the benefits of using MATLAB as a programming language for this project, based on MATLAB code, and how this differs from other programming languages, such as Python. The UML main class and types are also introduced. Generally, RTWP measures the total level of noise within the UMTS frequency band of any cell and captures uplink interference. A lower RTWP indicates a better channel. An unloaded network has an RTWP of around -105 dBm, whereas an RTWP of -95 dBm indicates some interference, and an RTWP of -85 dBm indicates strong interference [4]. Figure 2.1 illustrates the mean values for RTWP, uplink interference and severe uplink interference value.

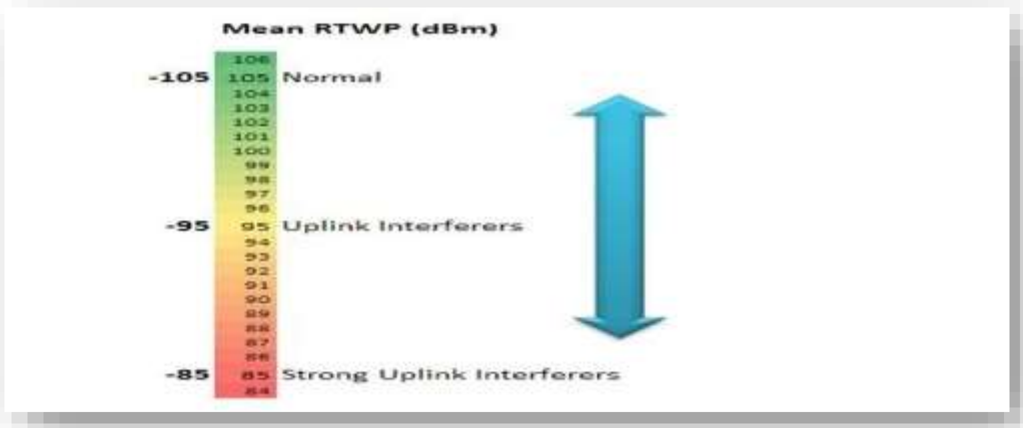


Figure 2.1 RTWP values in dB [5]

2.2 Overview of Received Total Wideband Power (RTWP)

RTWP is measured on the NodeB side, and it is the summation of radio links surrounding the NodeB plus the thermal noise power. Figure 2.2 illustrates this concept [18].

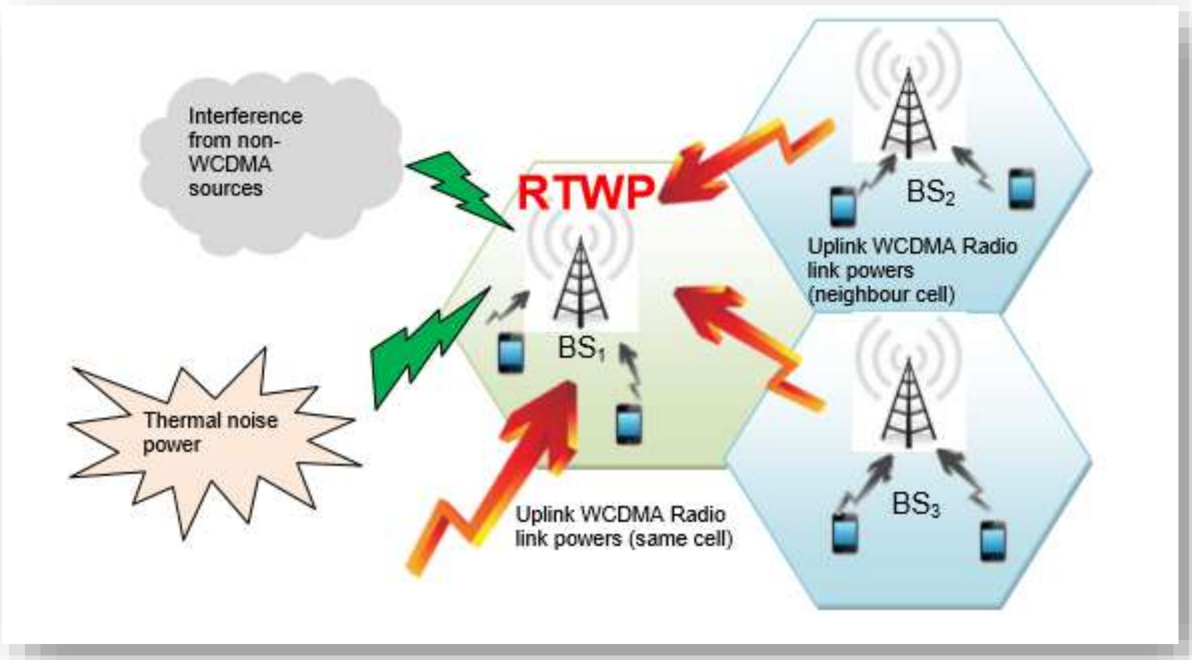


Figure 2.2 RTWP measurement [18]

Rise over Thermal in the equation below shows the proportion of the summation of the total uplink interference at NodeB side divided by the thermal noise [18]. It is an important factor in the design of wireless radio access networks. Equation 1 and Equation 2 show how Rise over Thermal and RTWP respectively are calculated:

$$RoT(t) = RTWP(t)/N(t) \qquad \textit{Equation (1)}$$

$$RTWP(t) = \sum_{k=1}^{k}(P_k(t) + I^N(t) + N(t) \qquad \textit{Equation (2)}$$

where:

$RoT(t)$ = Rise over Thermal

$RTWP(t)$ = Received Total Wideband Power

N(t) = thermal noise value that is recorded at the antenna terminal.

I (t) = the power radiated from the N neighboring cell surrounding the NodeB

$P_k(t)$= the external sources power that can be considered as external interference

Thermal Noise occurs in wireless communication systems when there is some minimum detection level that is determined by the noise floor, which relies mainly on the thermal noise power and the excess noise generated in the wireless communication system. Thermal noise is caused by the random movement of atoms in the material, and its spectrum is white [29].

The Power in (P_in) signal received by the antenna is amplified by the TMA, and a Remote Radio Unit (RRU) in a NodeB and then converted from a digital signal to an analog one. After that, the Power out (P_out) signal is the output. Therefore, RTWP indicates the power of a signal received by the antenna receiving port. The formula for calculating the Received RTWP is as follows:

$$RTWP = P_{in} = P_{out} - G \qquad \textit{Equation (3)}$$

In the preceding formula, (G) indicates the total gain of the receiving channel, namely the sum of the TMA gain and the NodeB gain. (G) is a constant value. Therefore, RTWP is measured at the NodeB and then reported to the Radio Network Controller (RNC) for access and congestion control. The RTWP on all antennas is measured at NodeB. In addition, the RTWP on each receiving channel in all cells is measured at the NodeB. Figure 2.3 shows the calculation that occurs only inside the receiving chain of NodeB [3].

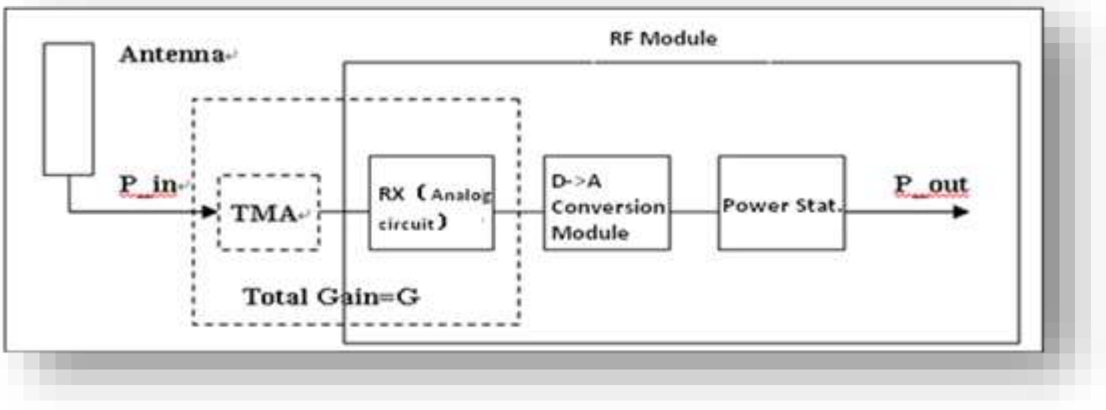


Figure 2.3 RTWP computation [3]

When no signal is input into NodeB, the RTWP measured in the NodeB equals the NodeB PN (Noise Power); the formula for calculating NodeB PN is as follows:

$$PN = KTB + NF \qquad \textit{Equation (4)}$$

where

K = Boltzmann constant

T = 290 K (room temperature)

B = RF carrier bandwidth (Hz) = 3.84 MHz

NF= Noise factor of the RF system

PN= Noise Power

Therefore, the NodeB PN is about -106 dBm at room temperature. The PN is affected by the analog circuit of the RF system (for example, component performance is affected by external factors, such as frequency and temperature) and changes due to the factor T. Therefore, a normal PN ranges from -108 dBm to -104 dBm [3]. Due to certain networking configurations, the NodeB PN rises in the following two scenarios:

- When the TMA or the line amplifier is used without the configuration of attenuating the RX channel.

- A NodeB uses the configuration of multiple RRUs.

The relevant terminology was discussed above, and a discussion on the congestion follows.

2.2.1 Congestion in Hardware And Software Can Cause the RTWP To Be Abnormal

Congestion occurs when the capacity of the system that has been allocated for a specific service is exceeded. In UMTS, power congestion can be observed in both uplink and downlink. When it is observed in downlink, it is called Transmit Code Power (TCP), and when it is observed in uplink, it is called RTWP. Channel Element (CE) is defined as the logical resource used for the baseband processing unit in the NodeB; the capacity of the services that can be provided by the NodeB is determined by CE congestion, which occurs when multiple users access different services and when the required CE is greater than the capacity of the system hardware [19]. An example of this scenario is the Dedicated Channel (DCH), which is the radio channel that is responsible for carrying voice services in a 3G network. Whenever this channel is increased in size, it will increase the CE utilization in the NodeB, since many users require voice services in the NodeB; RTWP will increase accordingly. Figure 2.4 shows the relationship between RTWP and DCH.

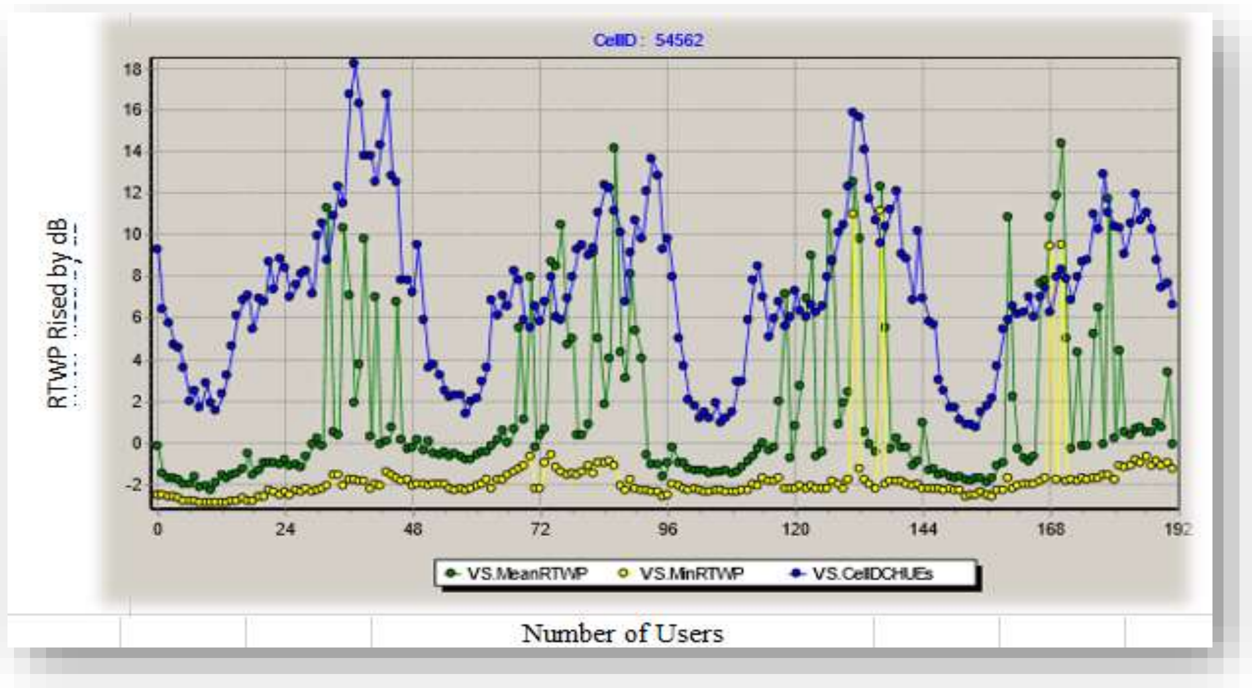


Figure 2.4 RTWP and DCH [3]

The green line indicates the MeanRTWP, the yellow line indicates the MinRTWP, and the blue line shows the CellDCHUEs (viz. the number of users in a cell). The figure above shows that the RTWP rises by several dB, as more users are served. The solution is to increase the CE resources in NodeB first by changing either the software or the hardware until the RTWP value has normalized. The TMA is a low noise amplifier designed to improve the coverage of the cell and capacity in wireless communication; it is installed near the antenna and it improves the uplink signal performance. One of the main reasons for using the TMA is to increase the coverage in the area where the received signal is weak and to reduce call interface and noise [25]. Therefore, any issue with this device could cause interference on the RAN side. Figure 2.5 shows where the TMA is installed in the NodeB.

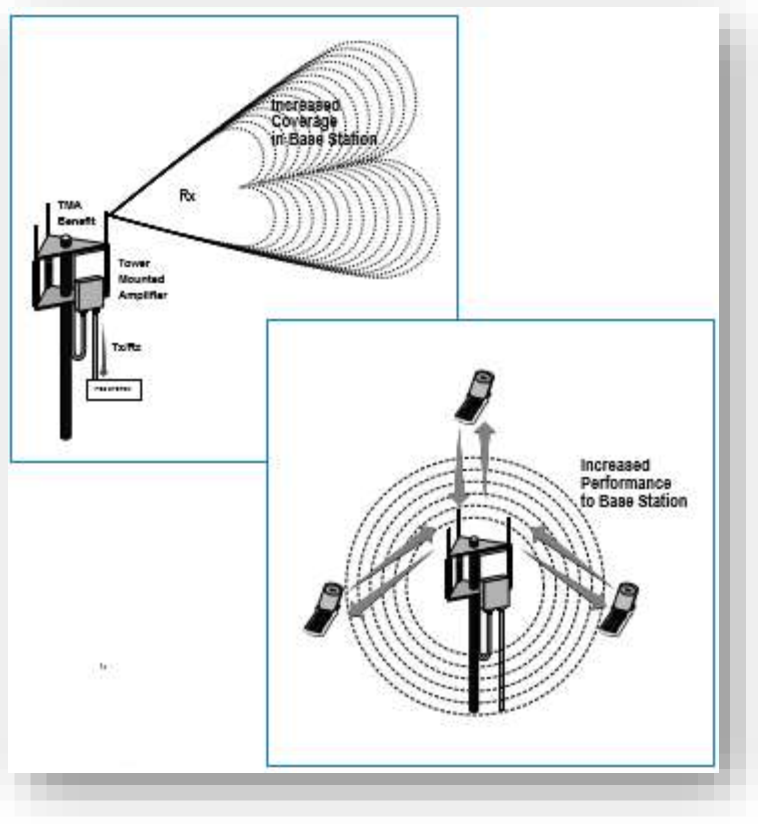


Figure 2.5 TMA installed on tower to improve coverage [26]

The TMA is widely used by operators to improve the uplink coverage in rural areas. The gain provided by the TMA is included in the link budget during radio planning; it can be a gain of 12dB or 24dB[38]. In the chapters 6 and 7 dealing with implementation and the testing results, the example of TMA 24 dB (48*0.5 TMA configuration) is used in the configuration.

2.2.2 Hardware Issue Can Cause RTWP Abnormality

An example of a hardware issue that can cause RTWP abnormality is crossed feeder connection, which generally occurs in the network during the installation of new sites or during routine maintenance for an antenna. The crossed feeder connection occurs when two or more sectors are inadvertently connected incorrectly [20]. Figure 2.6 shows the impact of the crossed feeder connection on the RTWP in both sectors.

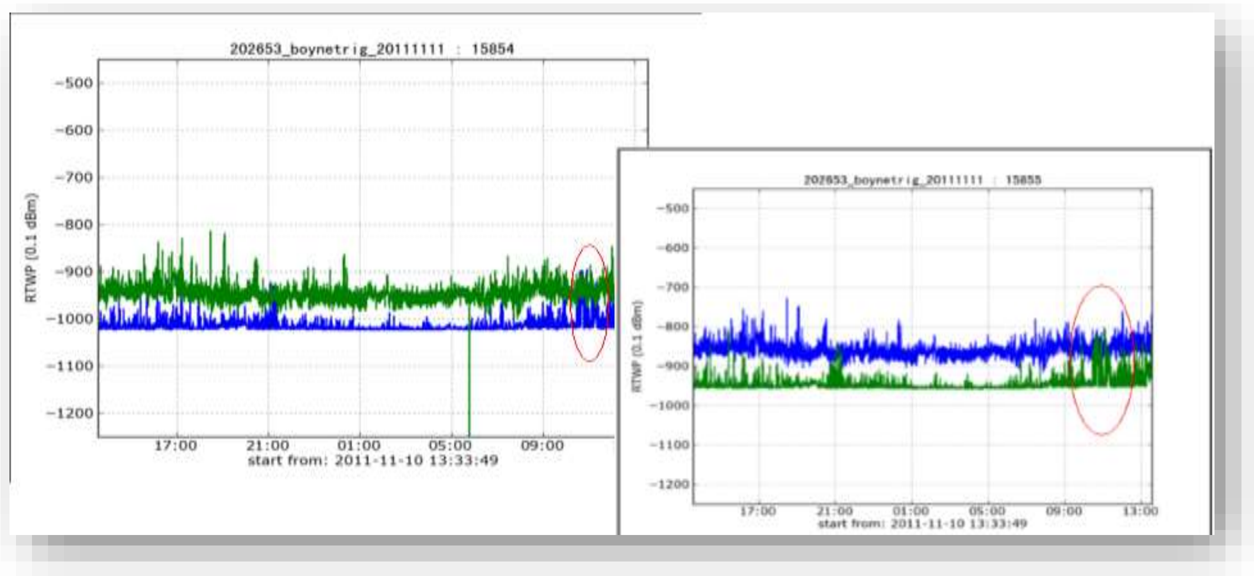


Figure 2.6 RTWP issues when crossed feeder connection exists between two sectors [3]

15854 is sector A and 15855 is sector B; each sector is represented by two values – main and diversity (green and blue line respectively). In the normal scenario, when there are no crossed feeder between two sectors, the RTWP value is represented by the main and the diversity values (green and blue lines in the first graph) in both sectors; however, when crossed feeder connections are observed between the two sectors, the main and diversity values swap, as can be seen from the blue and green lines in the second graph.

2.2.3 RF Interference Causes Abnormal RTWP

The RAN generally experiences interference from other wireless systems, which is why the

Telecommunication Regulatory Authority (TRA) assigns a specific frequency and guard band so that each operator can access its allocated frequency without interfering with another operator [6].

Figure 2.7 RTWP errors reported when interference is observed [6]

Figure 2.7 above is an example of a mobile network operator in Taiwan who suffered from interference in specific areas. In first graph, multiple-peak interference can be observed in two uplink (UL) channels, causing high RTWP. Figure 2.8 illustrates a high RTWP being observed in one site, whereas interference affects the area covered by the site. It shows that two UL bands were affected. Once this issue is triggered, the mobile telecom operator sends a drive test team to the target site and conducts several tests to locate the sources of interference, using a spectrum analyser or interference advisor [6], as shown in Figure 2.8.

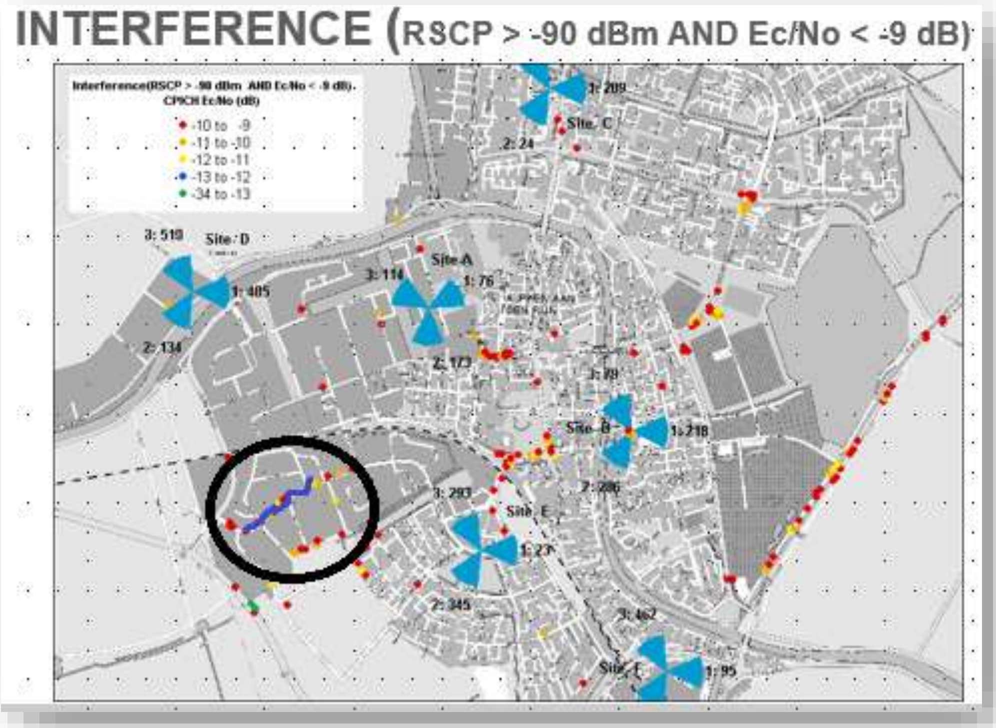


Figure 2.8 Problem area with high RTWP and test area [39]

The team conducted a brief manual interference hunt by driving around the area; they successfully located the source of the interference the blue area in Figure 2.8: is an example for drive test map to locate the sources of the interferenece it was an RF repeater hanging on a wall. In Figure 2.9, the team contacted the owner of the repeater, and as soon as it was turned off, the interference was eliminated [9], as shown in Figure 2.10.

Figure 2.9 Repeater that caused interference [6]

Figure 2.10 RTWP before and after turning off the repeater [6]

2.3 How the Operation Team Repaired RTWP in a Live Network

Most mobile telecommunication companies follow some sort of process to identify and overcome RTWP issues in a RAN. Generally, once the alarm is reported to the Operation Maintenance Centre (OMC) or once some degradation in KPI has been observed that is related to RTWP, the support team will check the configuration and perform some parameter changes or reset the board in NodeB. After confirming that the configuration is completely checked, the field maintenance team will check the hardware and make some changes in the antenna direction, such as azimuth or inclination. If no improvement is observed, the issue is escalated to the RF team, which conducts further tests and identifies the source of interference. Generally, the RF team uses a spectrum analyser or frequency hunter to identify the sources of interference.

2.4 Interference Measurement Procedure

Once the interference is observed in the receiving system, a spectrum analyser or interference hunter can be used to emphasize the presence of wireless signals in the operated frequency band of the mobile network. Figure 2.11 a below shows the device and how it is used by an RF engineer. The directional antenna determines the direction of the sources of interference; sometimes the RF engineer can use an omni-directional antenna [7]. Figure 2.11 b shows a spectrum analyser that measures the magnitude of an input signal versus the frequency within the full frequency range of the instrument. The primary use of these instruments is to measure the power of the spectrum of both known and unknown signals. The display of a spectrum analyser displays the frequency on the horizontal axis and the amplitude on the vertical axis. It shows the relative levels of signals on different frequencies within the range of the particular sweep or scan.

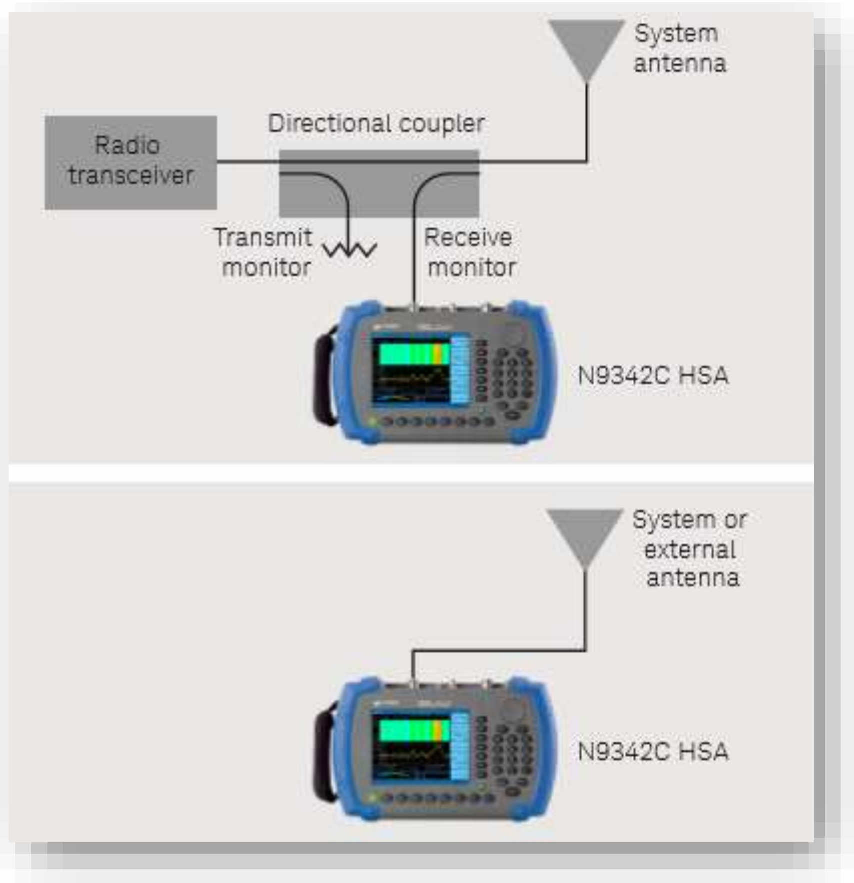


Figure 2.11 a Spectrum analyser configurations for measuring wireless interference [7]

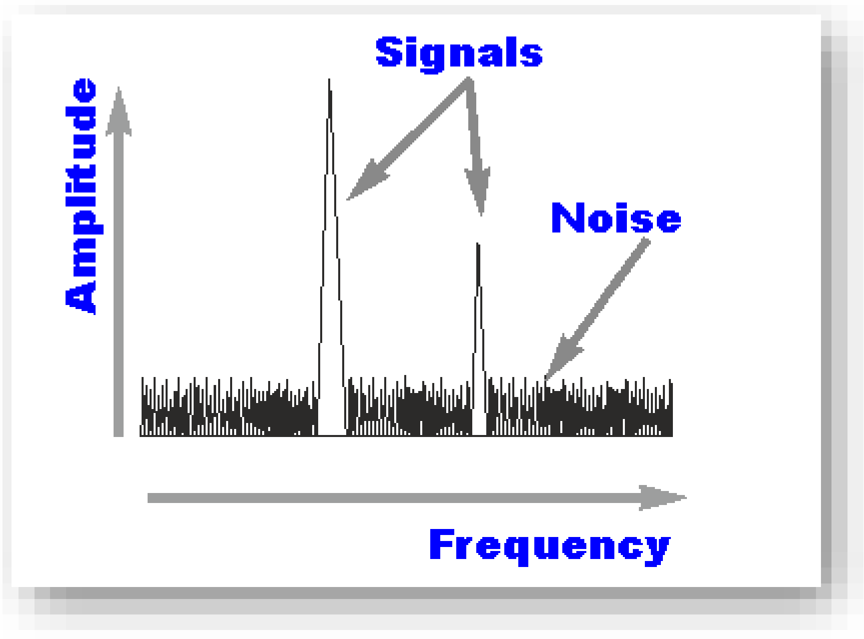


Figure 2.11 b Spectrum analyser signal representation [7]

2.5 Network Management

This part discusses how the NOC responds to any alarm in the network, especially RTWP alarms, and the tools used in the NOC to resolve issues in live networks.

2.5.1 Network Operation Centre (NOC)

The telecommunication operator relies on the NOC to monitor their networks, alarm diagnoses, remote troubleshooting, site information database, traffic monitor and service provision [21]. Telecom operators can operate either a centralized NOC or a decentralized NOC, depending on their infrastructure, geography and services [21]. Engineers in the NOC manage and control the network 24/7 to ensure the stability of the running network and the delivery of high quality services to network subscribers. The responsibilities of the NOC can be summarized as:

- Monitoring the network;
- Handling any issue related to the nodes in the network;
- Interacting with other functional departments, such as customer care, other operators NOC and public electricity suppliers.

2.5.2 Organization of the NOC

The NOC consists mainly of four parts, viz. the Front Office (FO), the Back Office (BO), Field Maintenance (FM), and Radio Network Optimization (RNO). The FO personnel is available 24/7 and can do some first-line troubleshooting, the BO is responsible for the second-line operation and supports the FM team, and it interacts with Research and Development (R&D) in case further troubleshooting is necessary. RNO is responsible for regulating the RAN and ensuring that the designed and planned network is implemented according to certain criteria and that the quality of service delivered and resources are enhanced and fully utilized. Mobile telecommunication operators rely on all four of these parts to run their networks in an efficient and economical way. Figure 2.12 sets out the normal procedure followed by a mobile telecommunication operator to solve RTWP issues. Figure 2.13 illustrates an organization for NOC, the Oman Telecommunication Company in the Sultanate of Oman, as a good example for explaining how the NOC operates in real life. The organization can be grouped into four levels: O&M Director, NOC Manager, Engineers Manager, and Engineers.

One of the main responsibilities of the NOC is to achieve the target SLA for alarm clearing, which is an indicator of the Quality of Service (QoS) for services that are delivered to network subscribers. Figure 2.14 shows an example of a NodeB SLA, which is an agreement between the mobile operator and the TRA regarding service disruption on the NodeB level. In addition to that, NodeB is classified according to geographical location, for instance, whether it is a Very Important (VIP) site. A node that is serving a highway or a rural area can be considered as a not-Very Important (VIP), but a NodeB that covers an area, such as an airport, hospital or a market can be considered as a VIP site.

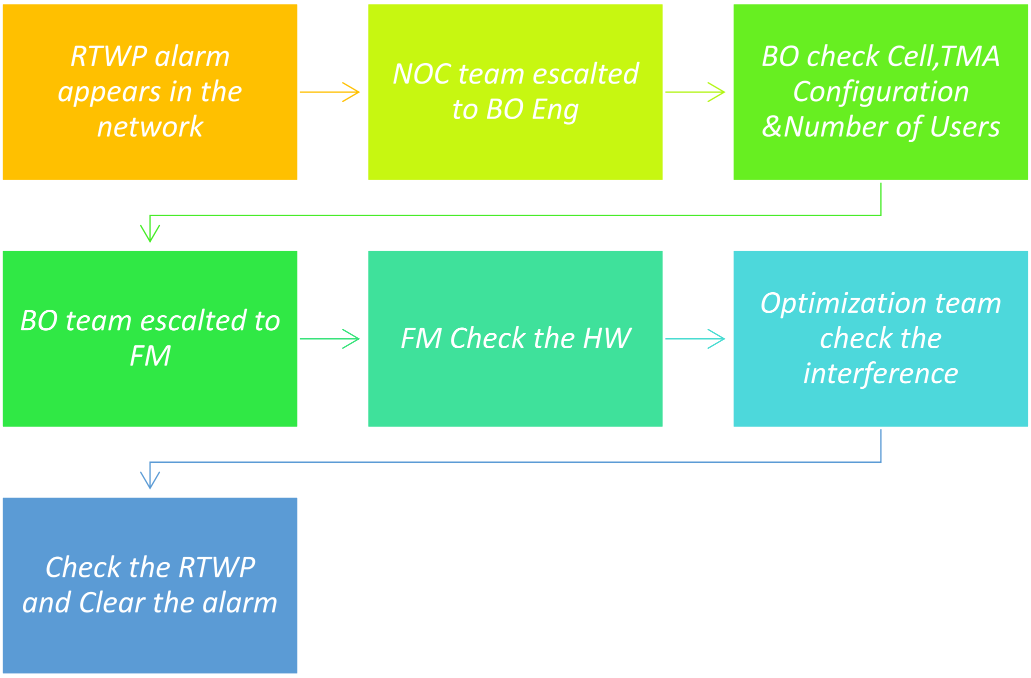


Figure 2.12 RTWP troubleshooting chain

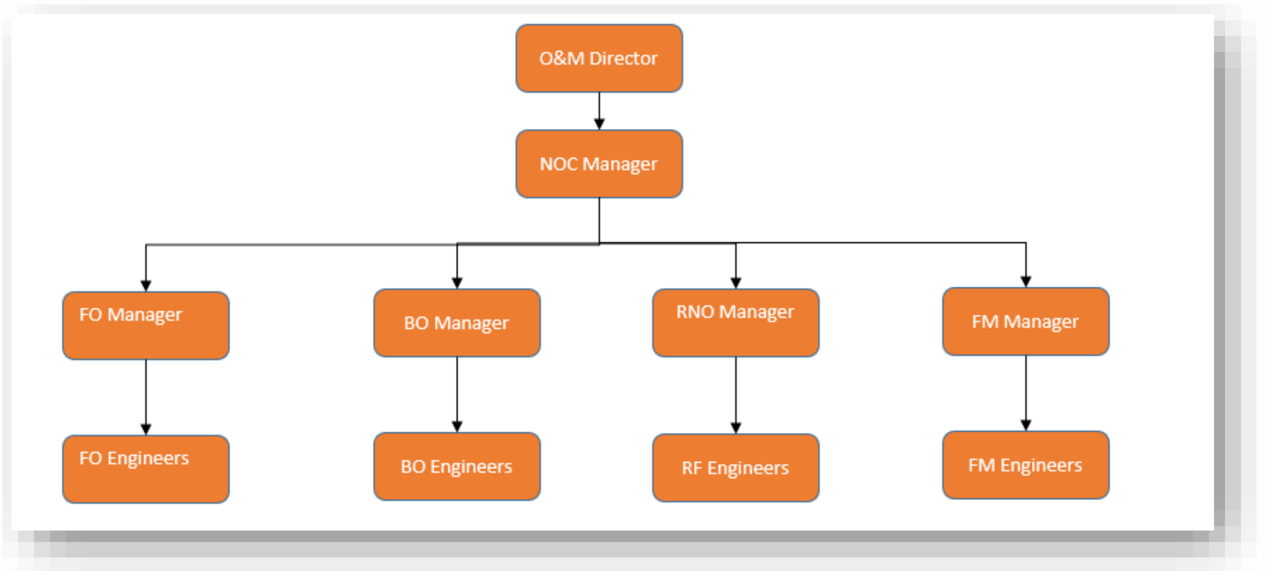
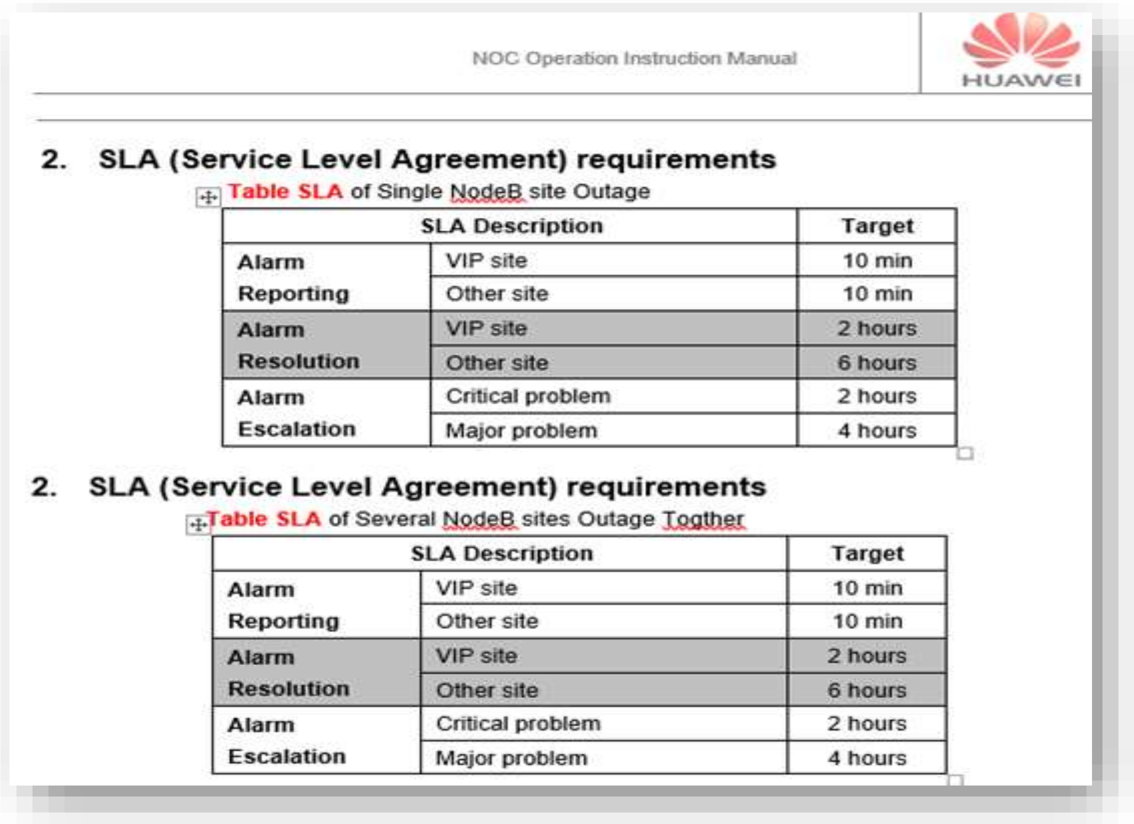


Figure 2.13 NOC Organization

The acronyms for the components of the NOC organization in Figure 2.13 above are listed in Table 2.1

Table 2.1 Acronyms for components of the NOC organization

Acronyms	Meaning
OMC	Operation Maintenance Centre
NOC	Network Operation Centre
FO	Front Office
BO	Back Office
RNP&O	Radio Network Planning & Optimization
RF	Radio Frequency



NOC Operation Instruction Manual

HUAWEI

2. SLA (Service Level Agreement) requirements

Table SLA of Single NodeB site Outage

SLA Description		Target
Alarm Reporting	VIP site	10 min
	Other site	10 min
Alarm Resolution	VIP site	2 hours
	Other site	6 hours
Alarm Escalation	Critical problem	2 hours
	Major problem	4 hours

2. SLA (Service Level Agreement) requirements

Table SLA of Several NodeB sites Outage Togther

SLA Description		Target
Alarm Reporting	VIP site	10 min
	Other site	10 min
Alarm Resolution	VIP site	2 hours
	Other site	6 hours
Alarm Escalation	Critical problem	2 hours
	Major problem	4 hours

Figure 2.14 NodeB SLA [2]

2.6 Tools Used in the NOC

The NOC uses many tools to run the network as efficiently and smoothly as possible, and to diagnose issues from a transport point of view, such as testing the connectivity, determining if the location is reached, or ascertaining if the device is working. Monitoring the background, which collects the events can also initiate their own probe and record the output in scheduled fashion [33], as shown in Table 2.2.

Table 2.2 Transport tool used in NOC

Tool Name	**Function**
ping utility	To test the connectivity of device and ensure reachability
Traceroute	Provides round-trip information between a source and destination IP address on a network
SNMP monitoring	Network interface is up or down
NetFlow	To monitor network congestion
SSH	For Telnet and remote log in

From the radio point of view, the NOC uses tools for capacity checking and traffic forecasting, cell planning and some Geographical Interface System (GIS) applications. Table 2.3 lists some of the tools used on the radio side.

Table 2.3 Radio tools used in NOC

Tool Name	**Function**
TEMSE	Field measurement
MapInfo	Field measurement analysis

2.7 UMTS Architecture

The UMTS architecture is based on the 3rd Generation Partnership Project (3GGP); the network can be divided into three different parts, viz. the User Equipment domain (UE), the UMTS Terrestrial Radio Access Network (UTRAN) and the Core Network (CN) part (see Figure 2.15). These are discussed below.

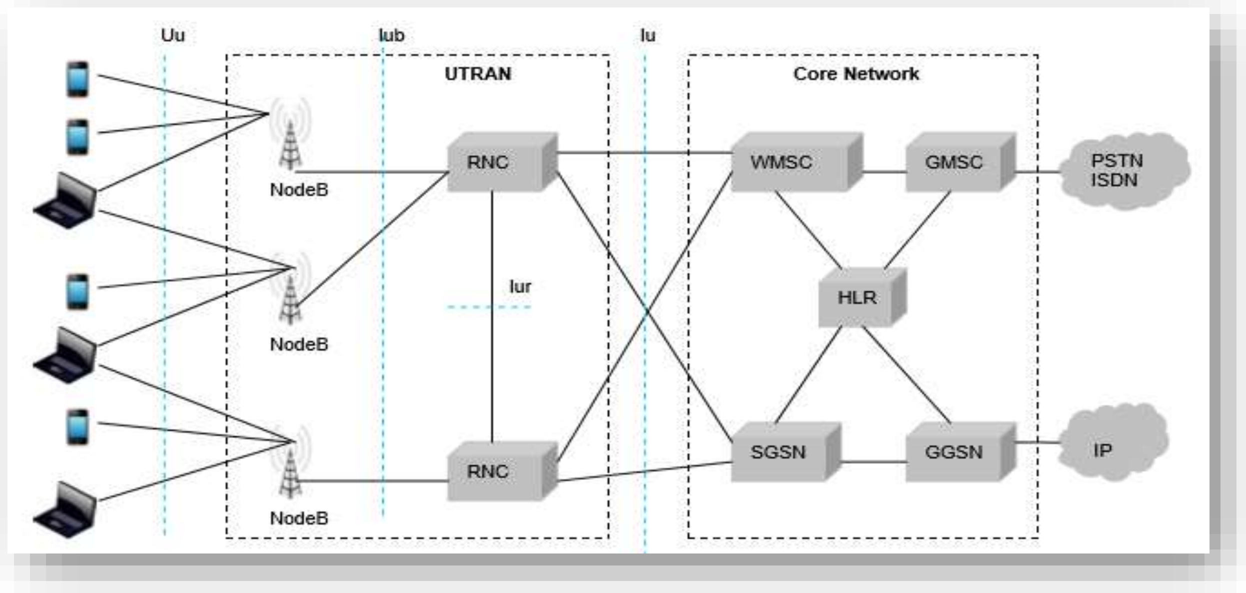


Figure 2.15 UMTS Architecture [20]

UE: User Equipment, which consists of the Mobile Terminal and the Sim card, is the device used by the subscriber to subscribe to the UMTS network and to receive the services offered by the UMTS [10].

NodeB: NodeB is a physical layer device that provides radio coverage, which allows the UE to access the network. NodeB provides soft handover, softer handover and hard handoff to the UE through the UU interface, and NodeB is connected to the Radio Access Network (RNC) through the IuB interface [10].

RNC: The RNC commands the NodeB connected to it and the radio resources provided to the NodeB, performing various tasks, such as encryption, managing the handoff and providing signaling to the UE [10]. The RNC is connected to the core network through the IuCS interface and the IuPS interface, to provide voice and packet services respectively.

Some of the most important interfaces in UMTS are the following:

- Cu: the electronic interface between the SIM card and the mobile terminal
- UU: the radio interface used by UE to access the network
- Iub: the interface that connects NodeB with the RNC.

- Iucs: the interface that interconnects the Mobile Switching Centre (MSC)] with the RNC

- Iups: the interface that connects the Serving GPRS Support Node (SGSN) with the RNC.

- Iur: the interface that connects the RNC with other RNCs.

SGSN: The Serving GPRS Support Node tracks the location of individual UEs in order to perform various security functions and access control measures; more generally, it provides the UE with the necessary signal and user plan information to establish Packet Switch (PS) services. This node is connected to the RNC through the IuPS interface [10].

GSGSN: The Gateway GPRS Support Node supports the edge routing function of the packet switched GPRS. The GGSN performs the task of an IP router for external packet data networks to protect the integrity of the GPRS core network. Firewall and filtering functionality are also associated with the GGSN along with a billing function [10].

MSC: The Mobile Switching Centre (MSC) server handles the mobility management, including the tasks previously performed by the Visitor Location Register (VLR); one MSC server can control multiple Radio Network Subsystems (RNS). Some connection management subtasks are carried out by the Media Gateway (MGW). Implementing the connection management in a MSC server increases the functionality, as adding new servers can increase capacity [10].

HLR: Home Location Registration (HLR) is a database located in the user's home system that stores the master copy of the user's service profile. The HLR database also stores the UE location on the level of MSC and SGSN [10].

When RTWP alarms occurs in the network, they will affect most of the KPI in the above interfaces, such as the quality of voice call, and the download and upload data rate for packet services.

2.8 Self-Organizing Networks (SON)

Self-Organizing Networks (SON) are a set of functions for automatic configuration, optimization, diagnosing and healing of cellular networks. They are a major necessity in future mobile networks and operations, mainly due to the possible savings in (CAPEX) and (OPEX) [31].

2.8.1 SON Architectures

The SON algorithm can be running on an Operation and Maintenance (OAM) server such as the Huawei U2000 or it can be running on eNodeB; depending on the location, SON can be divided into three main architectures: centralized, distributed and hybrid [32].

Centralized SON: In centralized SON, the optimization algorithm is running on an OAM server; Figure 2.16 shows how the centralized SON works [32].

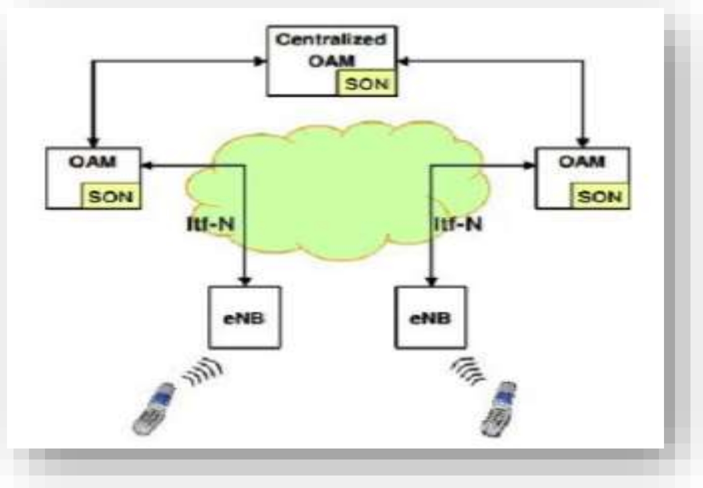


Figure 2.16 Centralised SON Architecture [32]

In **centralised SON** all the running functions are executed in the centralized OAM server where the Northbound Interface (Itf-N) is the interface between the Element Manager and the Network Manager. The problem with this approach is that, if the number of nodes is increased, a great deal of computation is required, which makes it difficult to scale up the system [32].

Distributed SON: In distributed SON, the optimization algorithm is running on eNodeB instead of on the OAM server. The drawback of this architecture is that it will increase the deployment efforts, and that coordination is required between the Nodes to ensure that the whole network is optimized [32].

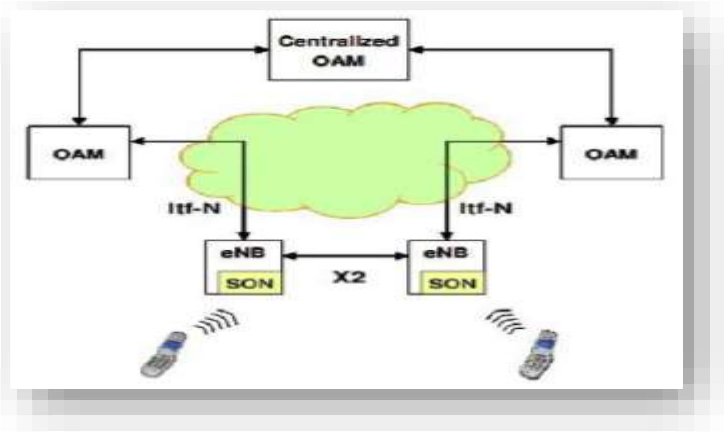


Figure 2.17 Distributed SON Architecture [32]

Hybrid SON: In Hybrid SON the optimization of a logarithm is partly distributed between the OAM server and eNodeB. Figure 2.18 shows an example of a hybrid SON: the simple optimization logarithm is executed on NodeB and the complex one is executed on the OAM server. The drawback of this approach is that much effort is required for deployment and interface expansion [32].

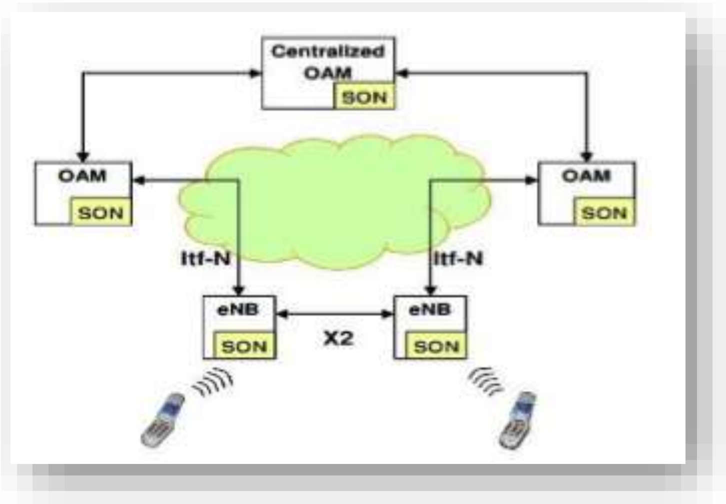


Figure 2.18 Hybrid SON Architecture [32]

2.8.2 Functionality of SON

The main functionality of SON includes: self-configuration, self-optimization and self-healing [32].

Self-configuration: Self-configuration is defined as a process where the eNodeB is being deployed and configured by means of automatic installation procedures, by getting the necessary configuration parameters from the OAM. This approach works during eNodeB implementation power on, and transmission resources will reach the eNodeB until the cells power on [32].

Self-optimization: Self-optimization is defined as a process where the UE and eNodeB measurement and performance measurement are used to auto-tune the network; this approach is used during network operation and gets the required measurement and performance statistic from both eNodeB and UE [32].

Self-healing: The Self-healing process is geared towards automatic detection and localization of faults; it applies the self-healing mechanism to solve several failures at the eNodeB cell level, configuration fallback and other software-related parameters [32].

The solution developed in this study can follow the self-healing principle or approach in future, since both deal with automatic detection and rectification of issues at the cell level. The main correlation between the GUI tool and SON is self-healing in terms of diagnosis at the RAN cell level. SON is primarily geared for LTE, but it can be extended to 3G/2G networks. [31] The developed GUI tool can be adjusted to solve many issues at the cell level, and it is not limited to interference; it can also solve load balance between the cells and the configuration of neighbouring cells. SON can also make eNodeB software or patch upgrades; the developed GUI tool can also be modified to enable upgrades for the NodeB.

2.9 Overview of UML

Unified Model language (UML) can be defined as a profile for testing paradigms and system modeling. It is the first part of the architecture [13]. According to Ukea and Thoolb [14], UML can be divided into the following:

- The structure diagram shows how the models or components within structure are interconnected with each other.
- The behaviour diagram shows the behaviour of the dynamic object within the system.

- The class diagram shows the structure of the system by presenting system classes, attributes, and methods, and the relation between the objects.
- The use case diagram shows the action between the user and system; it also describes the relation between use cases and users.
- The state machine diagram explains the action of the system when dealing with the external event in the form of machine state.
- The activity diagram describes the sequence of actions between different objects when activity takes place.
- The sequence diagram shows the interaction between objects while the task is being performed.
- The component diagram illustrates the components used to realise the functionality of the system.
- The deployment diagram sets out the architecture of the system from a deployment point of view.

The choice of the model depends on the application and the requirements; it is not necessary to use all of the models [14]. Therefore, in this research study, a sequence diagram was used to show the interaction of the user with the tool, and a class diagram was used to give an overview of the tools and how the objects' relation appears from a software point of view.

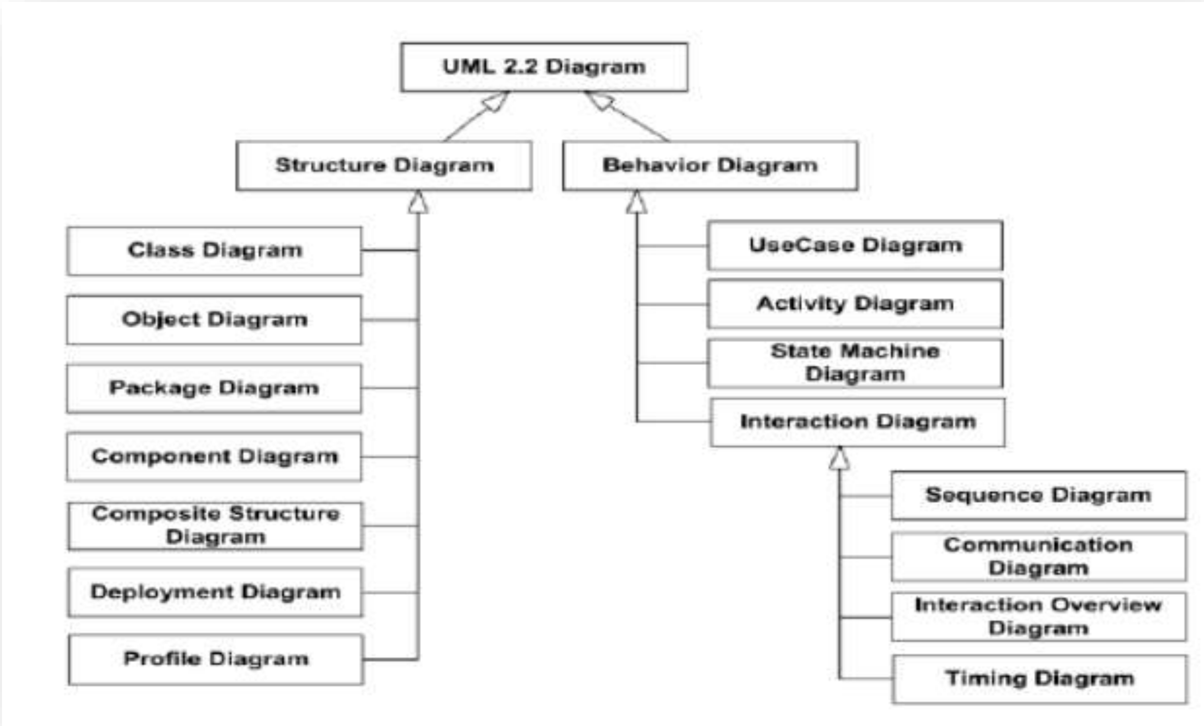


Figure 2.19 UML diagrams [14]

2.10 Overview of MATLAB and Python

MATLAB is a programming language that is widely used by engineers and data science analysts for computation. MATLAB can be installed on UNIX, Macintosh, Windows environments and can be installed on personal computers. One of the biggest advantages of MATLAB is that it can create data visualization for the user instead of using another computer program. MATLAB's easy to read data both before and after data analysis can give the user a broad view about the data and how it is presented [15].

Python is an object-oriented programming language with clear syntax and readability that can be easily accessed and used by most programmers and is thus in widespread use. Python can be installed on UNIX, Macintosh and Windows environments and on personal computers, and it can be used for both major and smaller projects [15].

Advantages of MATLAB over Python are:

- MATLAB is faster than Python for common technical computing tasks in statistics, engineering calculations, and data visualization [17].
- Algorithms can be run in parallel by changing for-loops into parallel for-loops [17].
- MATLAB contains a product called Simulink, which is a core part of the MATLAB package and not available in other programming languages [15].
- There is a very powerful tool box for GUI, which is the reason why it was chosen as the programming language for this research study.

Table 2.4 The for-loop differences between MATLAB and Python.

MATLAB	**Python**
started using for	started using for
counter equal to the array that is to be iterated over	counter set to be in the iteration array
loop has to be ended with an end statement	no end statement

2.11 Chapter Summary

This chapter reviewed in detail the background concepts relating to RTWP and RTWP alarms , and gave a brief introduction to a system that is designed to facilitate solving these issues. The chapter further recaps the theoretical background of RTWP and how it is measured; root causes were presented in this chapter and solutions proposed for each scenario where RTWP issues can be caused in a RAN. The current mechanism that is used by most mobile telecommunication operators, is the spectrum analyser: the chapter explains how it is used to identify the sources of interfaces in RAN. The TMA role in NodeB and how it is improving both the coverage and gain were discussed in this chapter, as well as the internal structure of the RX path in NodeB. The definition, structure and responsibilities of the NOC were set out, as were the SLA and how to deal with RTWP issues in live networks. An introduction to the UMTS architecture and the interconnection between nodes in the UMTS Network was provided. In addition, an introduction was given to SON architectures and types, which can be correlated to the work that has been carried out in this study, applying the same principles. Some discussion followed of UML and MATLAB, as both are used in this study to develop the GUI tool.

CHAPTER 3: METHODOLOGY

3.1 Introduction

This chapter will discuss the method of research that was used to design a GUI tool for troubleshooting the RTWP issue in a 3G RAN.

The term *methodology* is defined as "an approach, framework, process and structure that is designed to achieve a specific target", according to the Merriam Webster dictionary; furthermore "methodology is a body of methods, rules and postulates employed by a discipline. A procedure or set of procedures, the analysis of the principles or procedures of inquiry in a field'' [11].

In software development, many different methodologies are used to design software, such as Agile and the Waterfall processes [36]. In this research study, the Open Source Development model developed by Haddad for the Linux Foundation [12] was chosen as the method of developing the RTWP GUI tool. Figure 3.1 illustrates the method of development. In addition, an overview of the open source development methodology and the life cycle of the software product is given in Figure 3.2 and Figure 3.8 respectively.

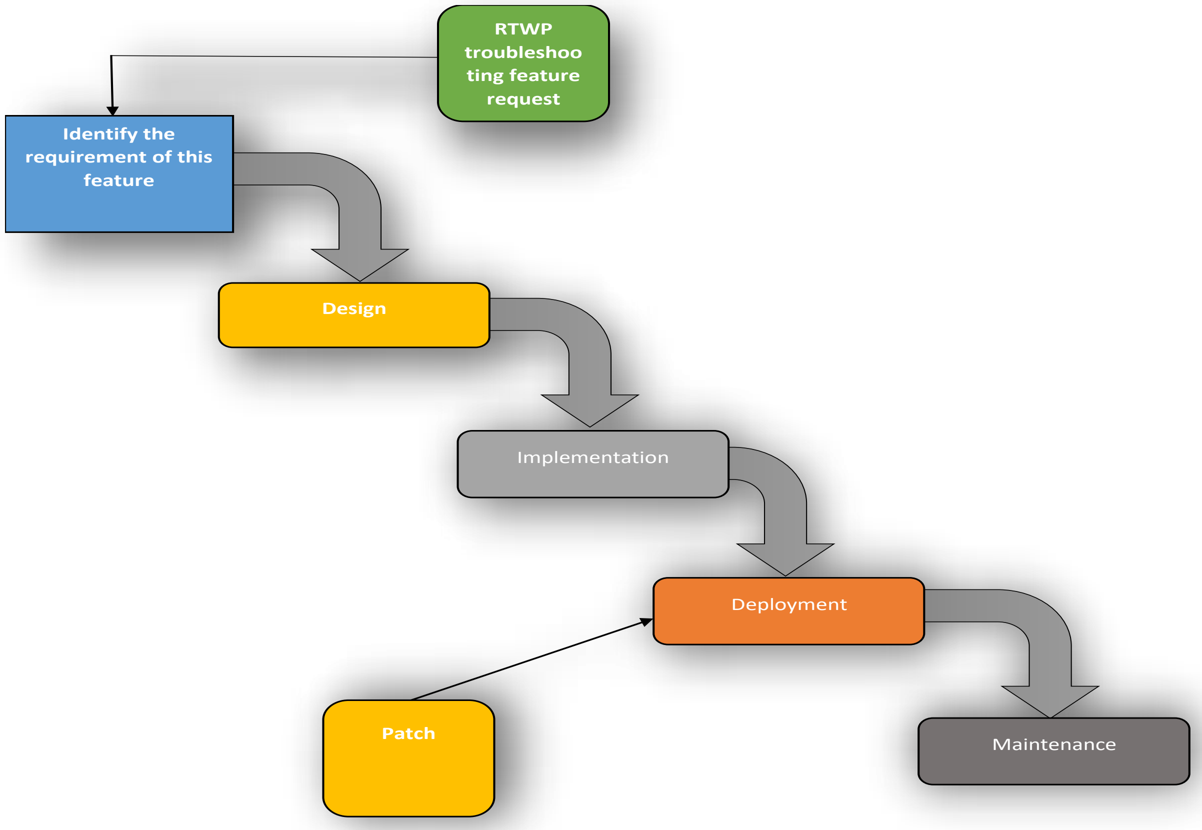


Figure 3.1 RTWP GUI development process

3.2 An Overview of the Open Source Process

The publicity of an open source development methodology in the software development community has increased during the last decade, and industry organizations have expressed their interest in using open source development methodologies [22]. There is a distinction between the new approach of the open source method of software development and the classical method of software development (understanding the requirements, design, implementation, test, releases, and maintenance) [23]. The open source development process is illustrated in Figure 3.2; the entire process is explained below:

- Mutual understanding of the requirements from the end user and the developer is necessary; for example, the aim of the project, and the components required to achieve the objective must be understood from both sides; the outcome of this step will be used for design and finding a solution [12].
- In the design stage, the idea of the project is translated into a prototype or running

software [12].

- In the implementation stage, the running software will be implemented and tested by the end user to provide feedback about the status of the project and thus to identify further improvement [12].
- In the deployment stage, the running software is integrated to an end user facility; integration will also be continuous with testing until the end user is satisfied with the outcome of the project [12].
- In the maintenance stage, the running software is running on the end user’s premises and it has been completely handed over to the end user, but the software developer will still receives feedback about the running status or any bug noticed during the operation and maintenance period; the developer is expected to be ready to provide support once an issue is triggered [12].
- In the patch stage, the developer and the end user will ensure the continuity of the project and its forward compatibility for further enhancement of their software product [12].

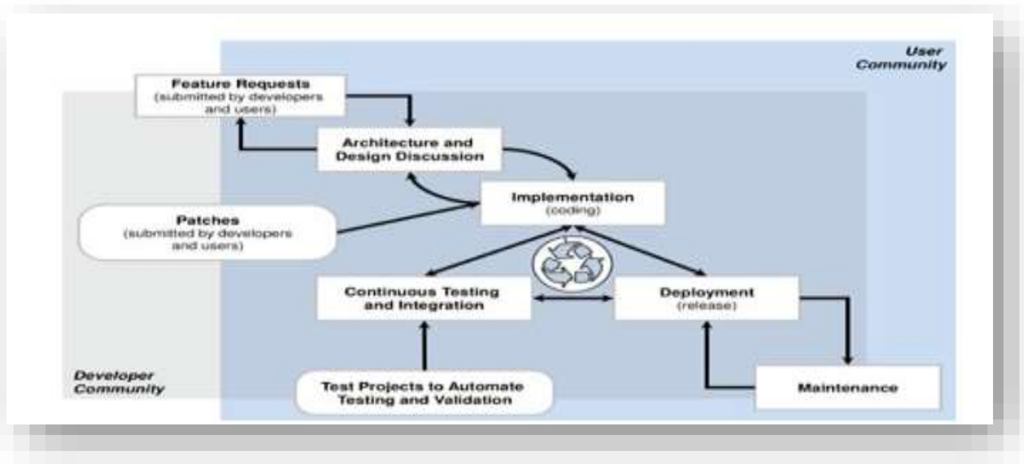


Figure 3.2 Open Source Methodology [12]

3.3 Stage One: Troubleshooting the RTWP Feature Request

RTWP issues generally occur daily in 3G networks; identifying the root cause of this issue needs much investigation, starting from the FO engineer, the BO engineer, RF optimization and the field maintenance team, and much collaboration work between them is needed. The root cause of RTWP alarms can be classified into three parts, regardless ofwhich vendor is used in the RAN (Huawei, Ericsson, ZTE, and Nokia):

- Wrong software configuration: a wrong setting or value in the NodeB Radio level configuration or board configuration can lead to this issue.
- Hardware issue: the board could be faulty, and/or the TMA could be faulty.
- External interference: from different sources could cause this issue.

Thus, to reduce the time and effort and improve the efficiency of troubleshooting RTWP errors, this author intends to design tools that will help mobile network operators to achieve their target SLAs and solve this issue within minimal timeframes. Once the tool has been designed, the mobile network operator can integrate this tool with their trouble ticket system or the Operation Switching Subsystem (OSS) to enhance their operation. In open source development, this is the first step in the software development process: the end user requests the software developer to understand the requirement of the requested model.

3.4 Stage Two: Identifying the Requirements of this Feature

In this stage, one must ensure a complete understanding between end user requirements and development requirements. The purpose of developing this tool, the resources needed to produce the tool, how this tool is to be developed and the expected outcomes have to be decided.

3.5 Stage Three: Design

After identifying the current shortcomings and gaps in troubleshooting RTWP alalrms, it emerged that many mobile network operators sometimes need more than one month to solve a RTWP issue on a single 3G site. It is necessary to design a solution and automate some of the processes that can improve the efficiency of troubleshooting and give clear guidelines for the end user about the root cause, and how to resolve a RTWP issue in the network. In the design stage, it is necessary to define the components, objects, and models of the tool and to link these with a block diagram. In addition to that, the designer must ensure that the end user requirements and the user interface tool are consistent and user-friendly. User experiences are the most important criteria and essential for creating a superior design. Documentation also helps in considering the design: the tool can provide help and give the user a clear description about the function of each button, as shown in Figure 3.3. Operation logs also help the administrator who is managing this tool to back-track and to use the documentation for any operation performed by this tool (see Figure 3.4).

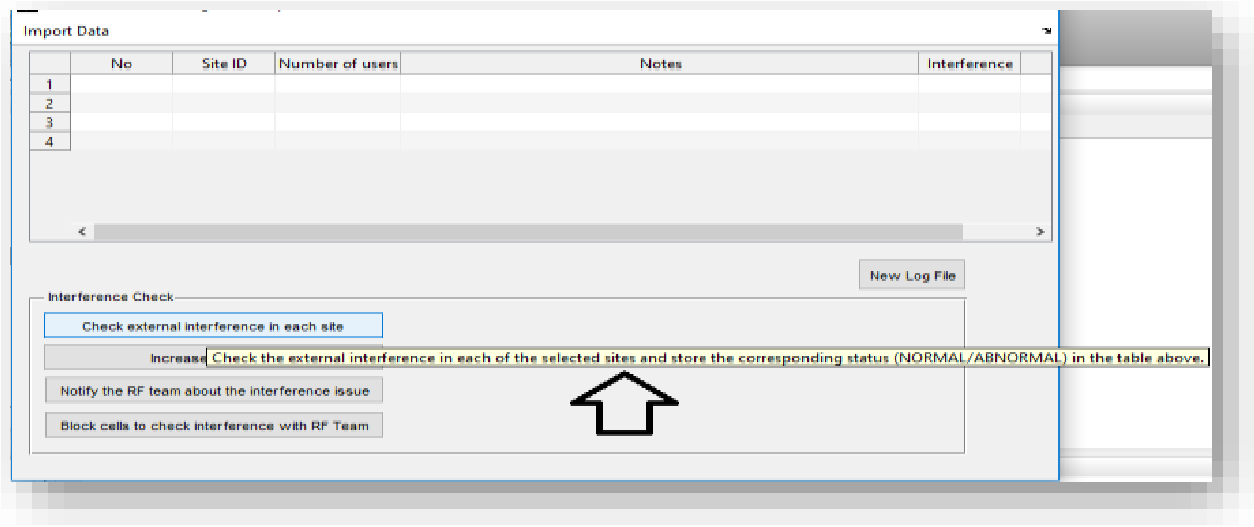


Figure 3.3 User help

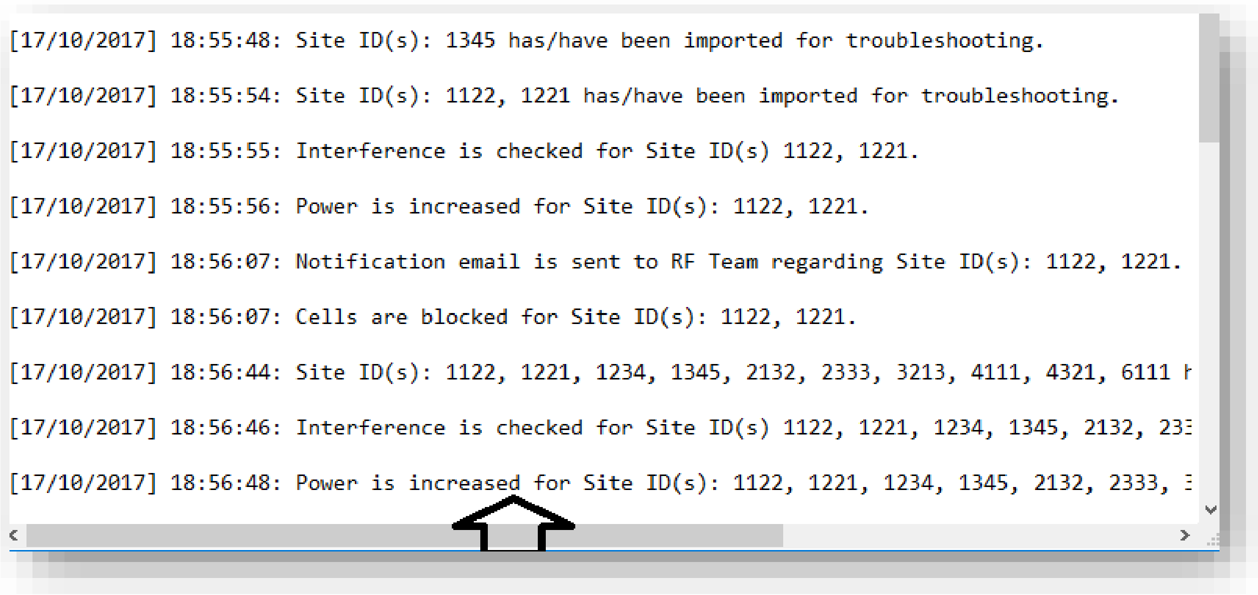


Figure 3.4 Operation log

3.6 Stage Four: Implementation

Implementation comprises carrying out, executing, and putting the design into practice. This stage must ensure that the code is constructed and tested and that the tool meets the requirements. Moreover, it must identify both the advantages and the drawbacks of the tool, and it must also identify customization options, if it is necessary. Construction of the user interface tool must use software, such as MATLAB, to achieve the optimum solution for meeting the requirements of the RTWP troubleshooting tool. During this stage, it is necessary to ensure the release of the tool, but before it is released to the deployment stage, testing and validation must be performed. It is also necessary to detect any issues and potential problems in the tool and to fix these before moving to the next stage. In addition, it is necessary to verify the tool and ensure that it satisfies the end user requirements before it moves to the deployment stage.

3.7 Stage Five: Deployment

This entails making the tool ready for use by the end user, by ensuring that it is running properly in an operational environment, after testing all the possible scenarios of a RTWP issue. Also, after reaching the maintenance stage of the tool, it must be determined if it needs any further enhancement or upgrade so that development can be continued to the patch stage. In this research, the tool was developed by using MATLAB R2014a; it runs on a laptop with the following specifications:

Table 3.1 Laptop specifications

Items	**Specification**
Operating System	Microsoft Windows 10
OS Type	x64-base
Processor	Intel(R) CPU 2.00GHz, 2000 Mhz, 2 Core(s), 4 Logical Processor(s)
Installed Physical Memory	RAM 4.00 GB

Another test has been conducted to evaluate the tools by using a different MATLAB version, such as R2017a at the University laboratory. During this test, many errors appeared due to differences in the MATLAB versions, especially with regard to the functions related to the GUI.

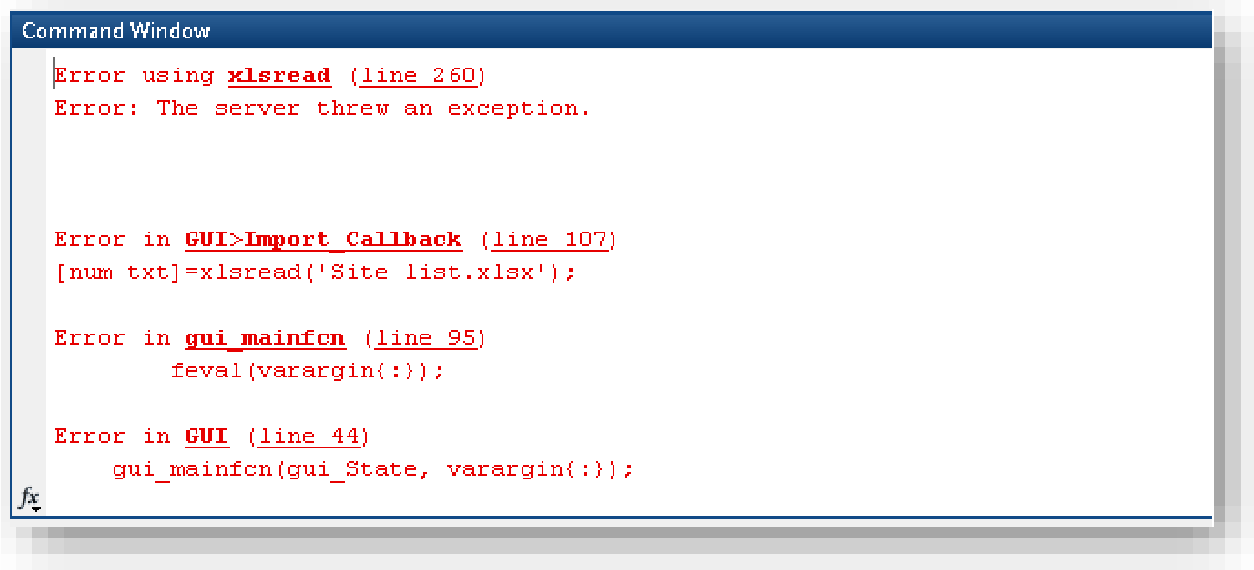


Figure 3.5 Errors as a result of using different MATLAB versions

3.8 Stage Six: Maintenance

By the time the tool is used by the end user in NOC, it has already passed all of the above stages, and been tested and verified by the end user. In the maintenance phase, the end user becomes familiar with the tool and its functionality (i.e. how it works). With time, the end users will gain much experience and can thus give feedback or share their opinions about it and they can advise on any further enhancement or modification that they regard as necessary.

3.9 Stage Seven: Patch

In case the tool needs any enhancement or future work, it can be upgraded to make any such further enhancement to it or to any software to counteract any bug noticed during the operation and maintenance stage. Generally, such patching will ensure the continuity of using the GUI tool and also ensure its forward compatibility. For example, if it is needed to modify the tool to support another vendor type, such as Ericsson NodeB, Nokia NodeB or ZTE NodeB, the developer just needs to search for the relevant Huawei commands, which are generated by the tool, and to read these with the tool as well, replacing them with a command in the suitable platform. If there is a need to replace the GUI tool to support Ericsson NodeB, in Huawei NodeB the command that checks the crossed feeder connection status is 'STR CROSFEEDTST', which means "Start testing crossed feeder connection status in the Huawei

NodeB’. On the other side ‘RXI>>’ means ‘Show cross connection in Ericsson NodeB’ as shown in Figure 3.6.

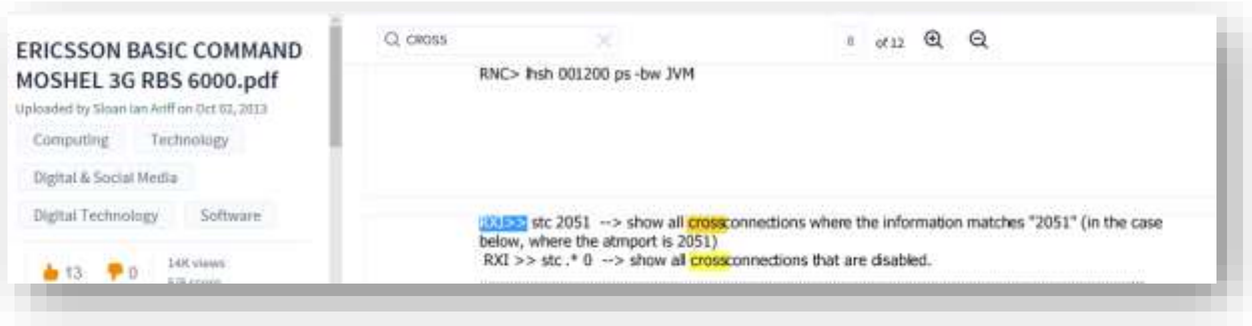


Figure 3.6 Ericsson NodeB Crossed feeder commands [24]

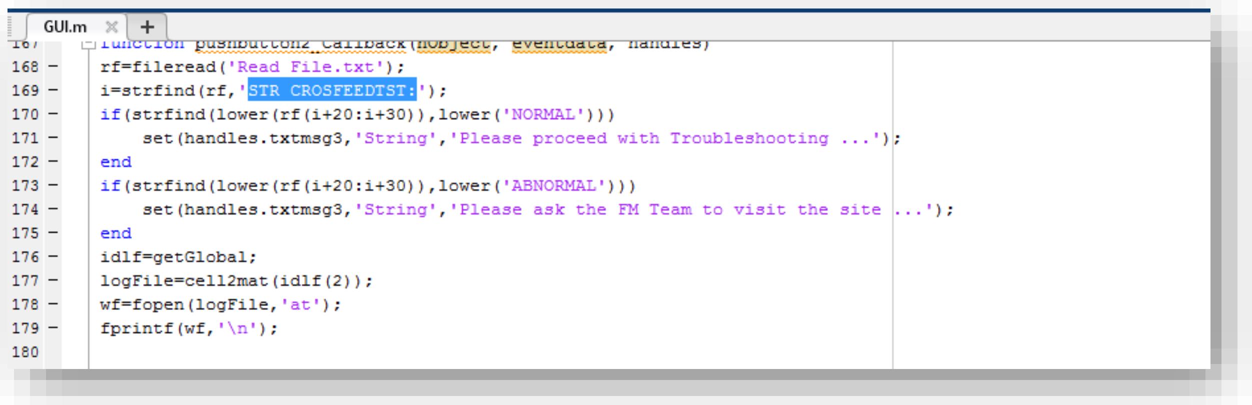


Figure 3.7 Crossed feeder command in the developed tool

In this case, the developer would need to go to MATLAB tool line 169 and replace ‘’STR CROSFEEDTST’’ with ’RXI>> ‘’, line 170 with ‘’ ENABLE’’ line 173 with ‘’DISABLE’’.

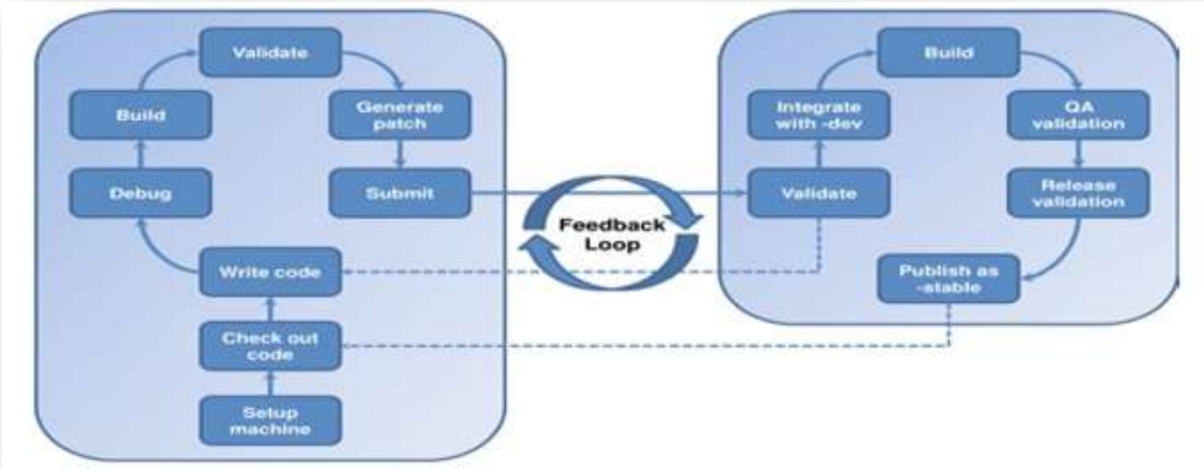


Figure 3.8 Patch life cycle driven by the developer and maintenance engineers [13]

The life cycle for any running software illustrated in Figure 3.8 reflects the patch life cycle driven by both the maintenance engineers and the developer's teams to ensure the continuity and stability of the running software. The maintenance team is responsible for submitting their request for adjustment and modification to the development team. They rely on the developer for developing a solution and must ensure that it satisfies the requirements of the end user. Another option is for the development team to develop a new feature to enhance the running software, which they would like to deploy. The developer can start communicating with the maintenance team about their solution and plan for implementing it in the running software.

3.10 Chapter Summary

This chapter provided the details of the development process for the GUI tool. It is based on the open source development process, which has special characteristics that allow faster development by continual testing, faster innovation and easy involvement of the various stakeholders (end users, developers, maintenance engineers, etc…). For this reason, it has been chosen as the best development process in this situation. Discussion followed with regard to the major elements for developing the RTWP troubleshooting GUI tool. Good understanding between the end user and the development community, and continuous assessment in each stage to guarantee the reliability and flexibility of the tool are required. In addition, the design used to run the tool based on MATLAB and the example presented about how to make further adjustments in the tool to support another vendor, such as Ericsson NodeB, are given.

Comparisons were presented between the traditional software development method and the open source model. The life cycle of the tool to be released and how it can be maintained in an optimum way to ensure continuity was also discussed.

CHAPTER 4: DESIGN

4.1 Introduction

This chapter will present the system diagram for the GUI tool that was designed to resolve a RTWP issue on both a single site and in multiple sites: it will be discussed how the user will interact with the GUI tool in both scenarios and how the GUI tool will check the NodeB configuration related to RTWP issues, and the expected outcome if the configuration is found to be right or wrong. Due to the complex nature of the software, it is difficult to understand the entirety of the system from a single viewpoint [13]; thus, following good practices, modeling techniques have been used to present different aspects of the system to explain the different parts more clearly. Accordingly, the Unified Modelling Language (UML) has been used to represent the different aspects of the system [13]. A range of UML diagram types has been presented in this chapter in the subsections that follow. Subsequently, the system implementation is presented in more detail to provide a fuller view of the operations. Findings in terms of system performance are investigated in respect of the different parts of the system. The modeling for the system is presented below, starting with sequence diagrams presented in the next subsection.

4.2 Use Case Diagram for Single-site Mode

The use case diagram shows the high-level user-system interaction that takes place when the tool is used. Figure 4.1 shows the main UML use case diagram for the GUI tool. The purpose of this diagram is to show the actions between the user and NodeB in single-site mode.

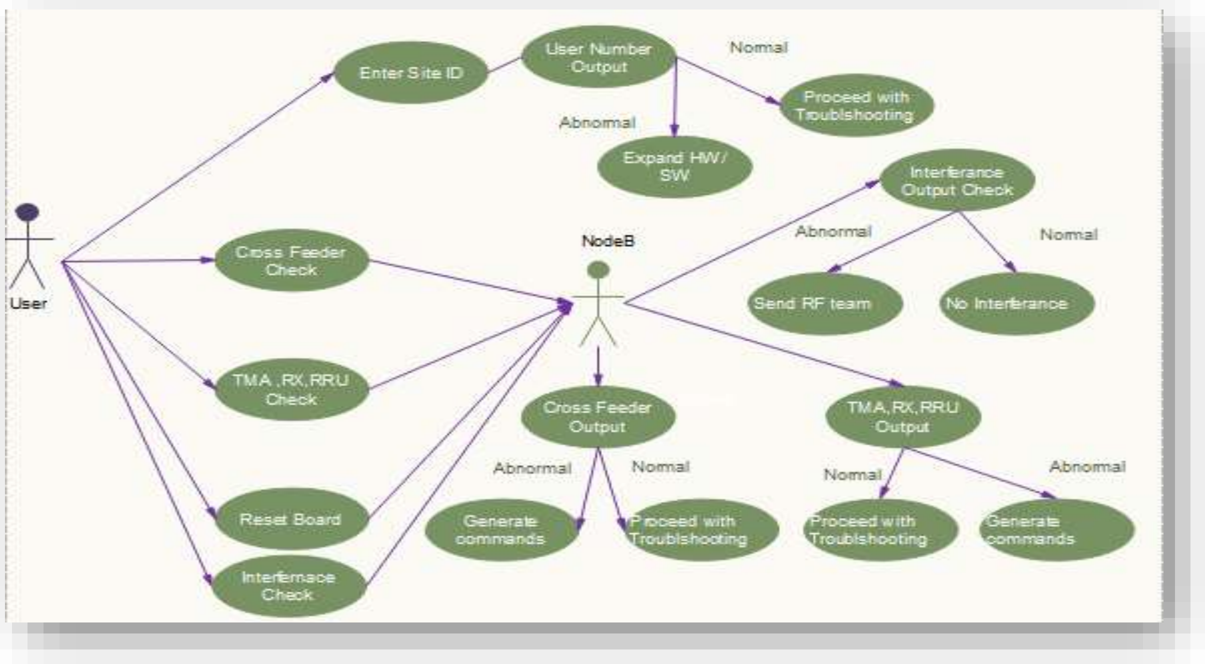


Figure 4.1 Use case diagram for tool in single-site mode

4.3 Use Case Diagram for Multi-site Mode

Figure 4.2 shows the UML use case diagram for the interaction between the user, the GUI tool, and NodeB in multi-site mode.

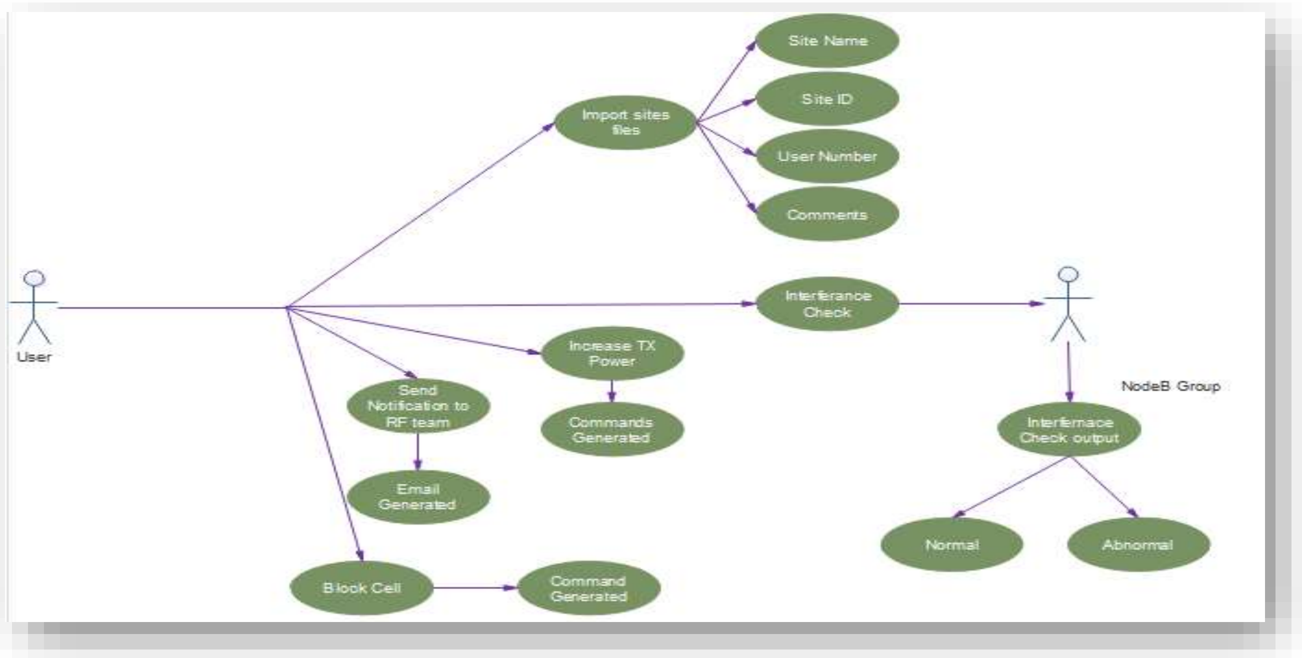


Figure 4.2 Use case diagram for tool in multi-site mode

4.4 Class Diagram for Single-site Mode

As indicated in the preceding sections, the tool needs to comprise a number of pieces. These pieces are illustrated in the class diagram in Figure 4.3. This class diagram also describes the structure of software system design for the GUI tool, emphasizing aspects of it, and decomposition (shown by the solid diamond relations) and aggregation (shown by the unfilled diamond relations). The user input site ID and output might be commands generated to fix issues, to issue notifications and to display the status of troubleshooting, such as number of users and RX branch configuration.

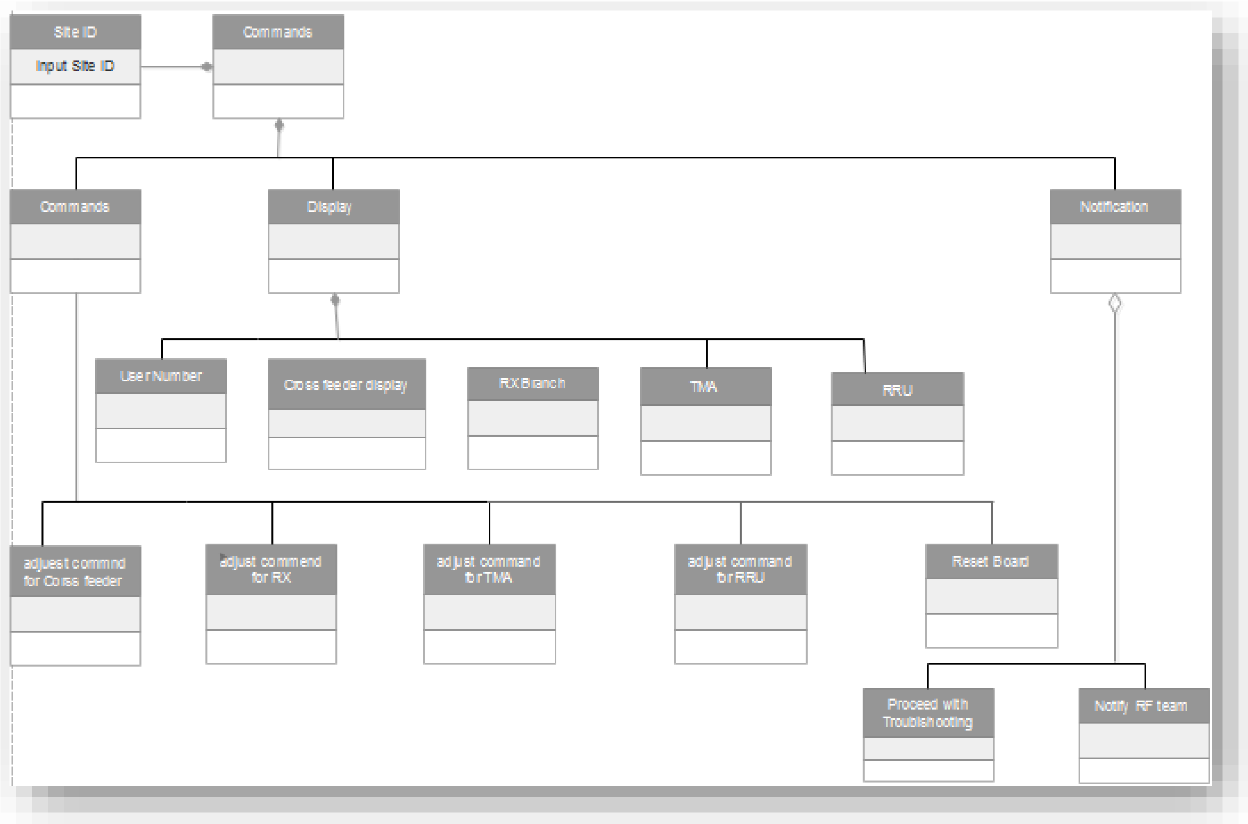


Figure 4.3 Class diagram for the tool in single-site mode

4.5 Class Diagram for Multi-site Mode

The class diagram in Figure 4.4 describes the structure of the software system design for the GUI tool operating in multi-site mode to troubleshoot problems across multiple sites. The user uploads the configuration files, and the tool displays the status of the configuration, sends notification or generates commands.

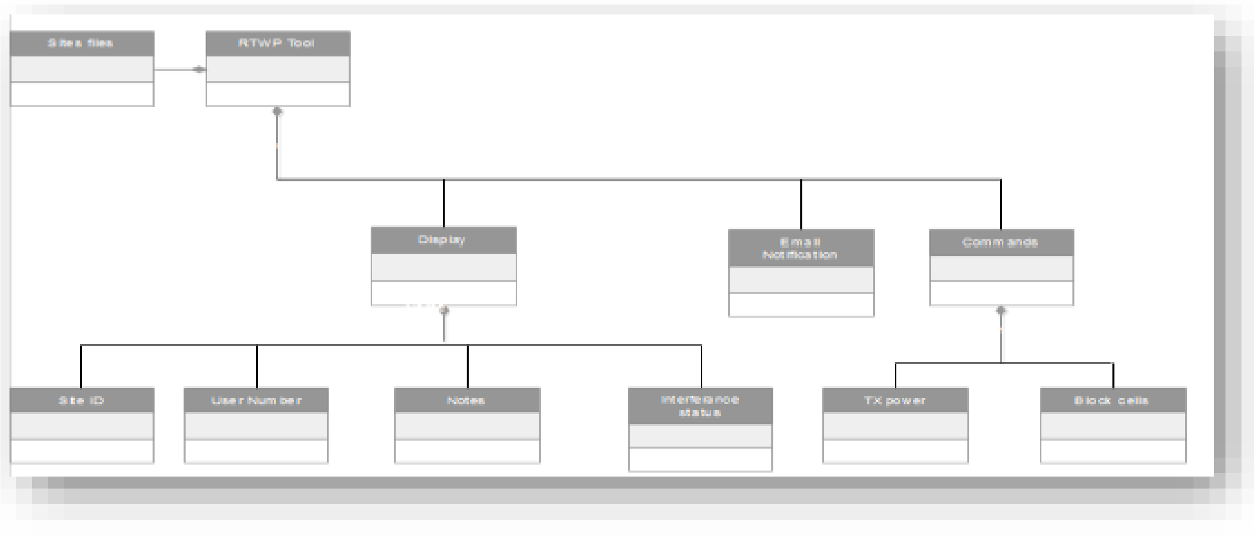


Figure 4.4 Class diagram for tool in multi-site mode

4.6 Sequence Diagram for Single-site Mode

The sequence diagram in Figure 4.6 illustrates the interaction between the user, the tool and NodeB to solve a RTWP issue in a single site. As stated in Section 2.9, a sequence diagram represents the interactions between objects while tasks are being carried out.

4.6.1 Action from the user and feedback from the tool

The following list of activities explains the user's interactions with the system, giving insight into actions that must be incorporated in the sequence diagram.

- The user inputs the 'site ID', which is affected by the RTWP problem.
- The tool search for the number of users that corresponds to the 'Site ID' that is currently erved by this site (in the live network, there is a performance counter that keeps track of the current number of users served by NodeB); this counter is simulated in the project by means of a constant value. The study author defined the threshold in this project as 90 users per second who can be served by Huawei NodeB. If the number of users is greater than 90 users, the tool will suggest expanding the hardware and software of the NodeB; if it is less than 90 users, then it will suggest proceeding with troubleshooting.
- The user requests the tool to check the crossed feeder connection in NodeB.

Examples of such crossed feeder configurations are shown in Figure 4.5 in the right side the impact of crossed feeders showed sector A3 appeard while it should be sector A2. Normal site operation should be like the left site scenario.

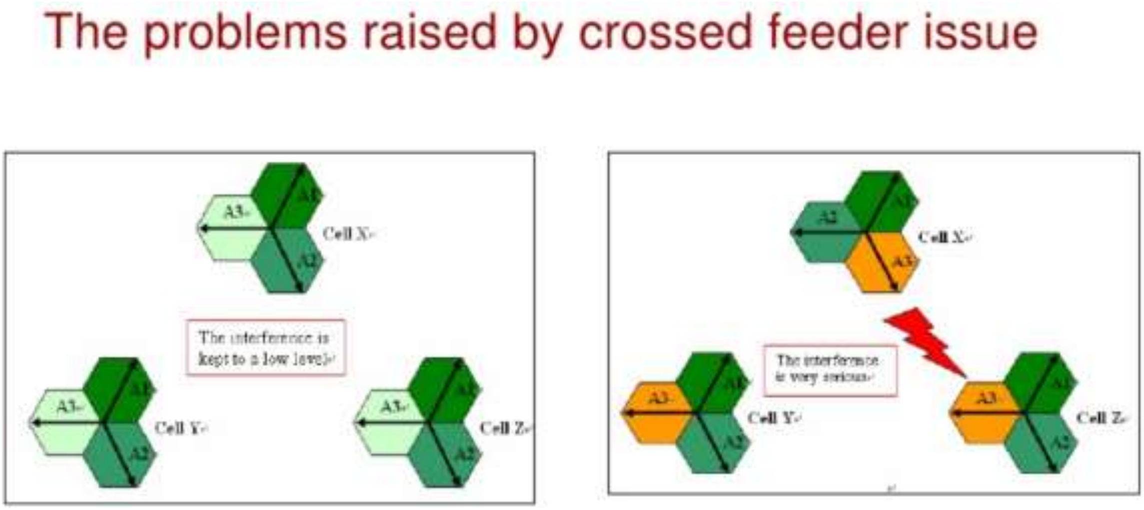


Figure 4.5 Crossed feeder problems [20]

In the cellular industry, ‘crossed feeder’ is a term widely used to describe the situation when two sectors in a single site are inadvertently connected the wrong way [20]: it can cause high RTWP. It generally means that the transmitter (TX) and receiver (RX) feeders are swapped, viz. that TX terminated at one point instead of RX and that RX terminated at a point instead of TX.

To solve the crossed feeder problem in the network, it is necessary for the field maintenance team to visit the site and fix the problem manually. During this process, to assist the maintenance team, the tool opens the NodeB configuration file and searches for status information concerning the Crossed feeder issue, before suggesting to the field maintenance team to visit the site for further troubleshooting, and giving details, such as the site ID and name.

- Once the user clicks on TMA Configuration Check, the tool checks the configuration for the RX branch, the TMA and the RRU to see if any wrong configuration settings are found in the Read file of the site; then the tool generates a correct command and the operation engineer should execute this command in the NodeB.

- Once the user clicks the Reset Board button, the tool will generate a reset command,

and execute this command on the relevant NodeB.

- Once the user clicks on Check Interference, the tool reads the file and checks the interference status in the configuration file. It reports to the user if the status is normal or abnormal. If it is abnormal, it will ask the user to send the RF team to the site.

The operations discussed in the text above have been represented in the sequence diagram in Figure 4.6, which describes the sequence of actions that the user needs to perform to interact with the tool in order to resolve the RTWP issue in a single site.

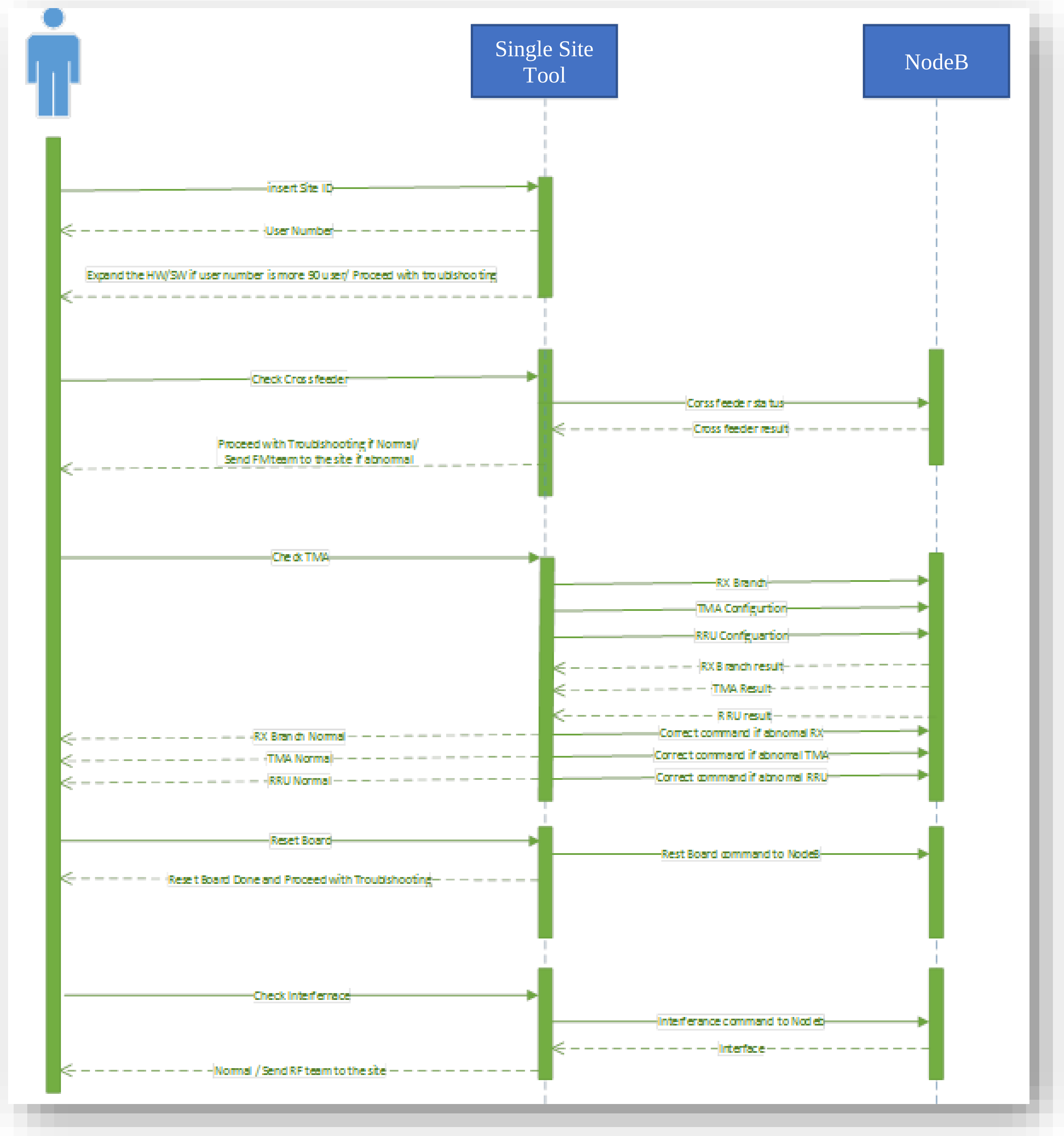


Figure 4.6 Sequence diagrams for tool in single site mode

4.7 Sequence Diagram for Multi-site Mode

The sequence diagram in Figure 4.7 shows the use of the tool for fixing an RTWP problem on multiple sites. The diagram shows the interactions between the user, the tool and a group of NodeB.

4.7.1 Action from user and feedback from tool in multi-site mode

The following set of actions are carried out by the user when interacting with the tool in multi-site mode:.

- The user imports the Read files of the sites that experienced the RTWP problem in the network; the tool will read the file and show the number of users served by the sites, the site ID and how it is served.

- The user will click on Interference Check and the tool will go to Read file and search for the interference command and show it to the user, i.e. whether it is normal or abnormal.

- The user will click on Increase TX Power and the tool will generate the command to increase the TX power in the relevant cell in the Write File.

- The user will click on Notify RF Team about the interference issue, and the tool will send an email to the RF team (each team has a communal group email address to which the tool sends the email) to inform them about this issue.

- Once the User clicks on Blocking Cell, the tool will generate commands to block the cells in the Write File so that the RF team can check the interference in the relevant sites.

The sequence diagram in Figure 4.7 describes the sequence of actions that the user needs to interact with the tool in order to resolve the RTWP issue across multi site .

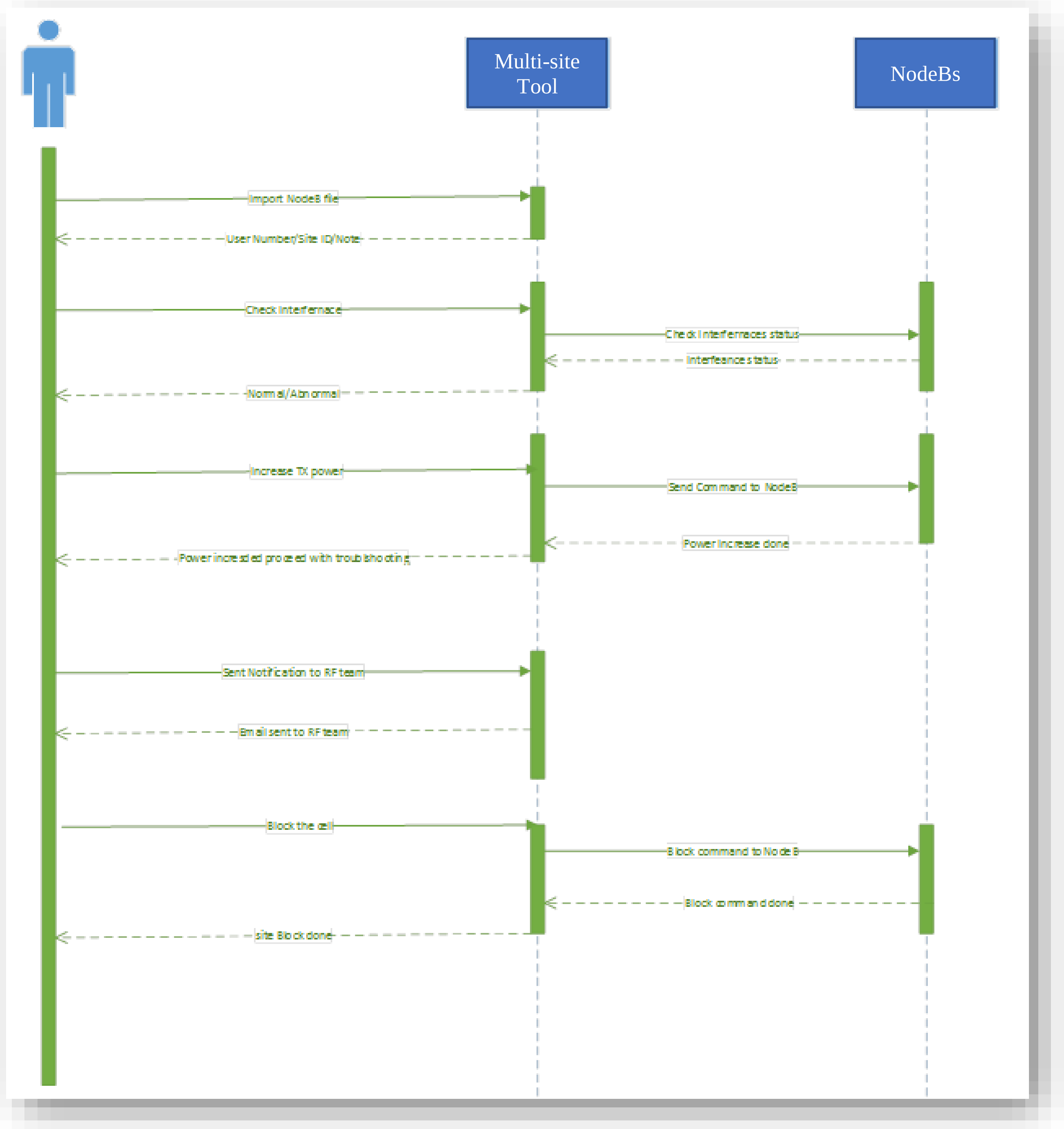


Figure 4.7 Sequence diagram for tool operating in multi-site mode

4.8 Activity Diagram for Single-site Mode

Figure 4.8 shows the UML activity diagram for the GUI tool. The purpose of this diagram is to show the sequence of actions between the different objects of the system when tasks take place in single-site mode.

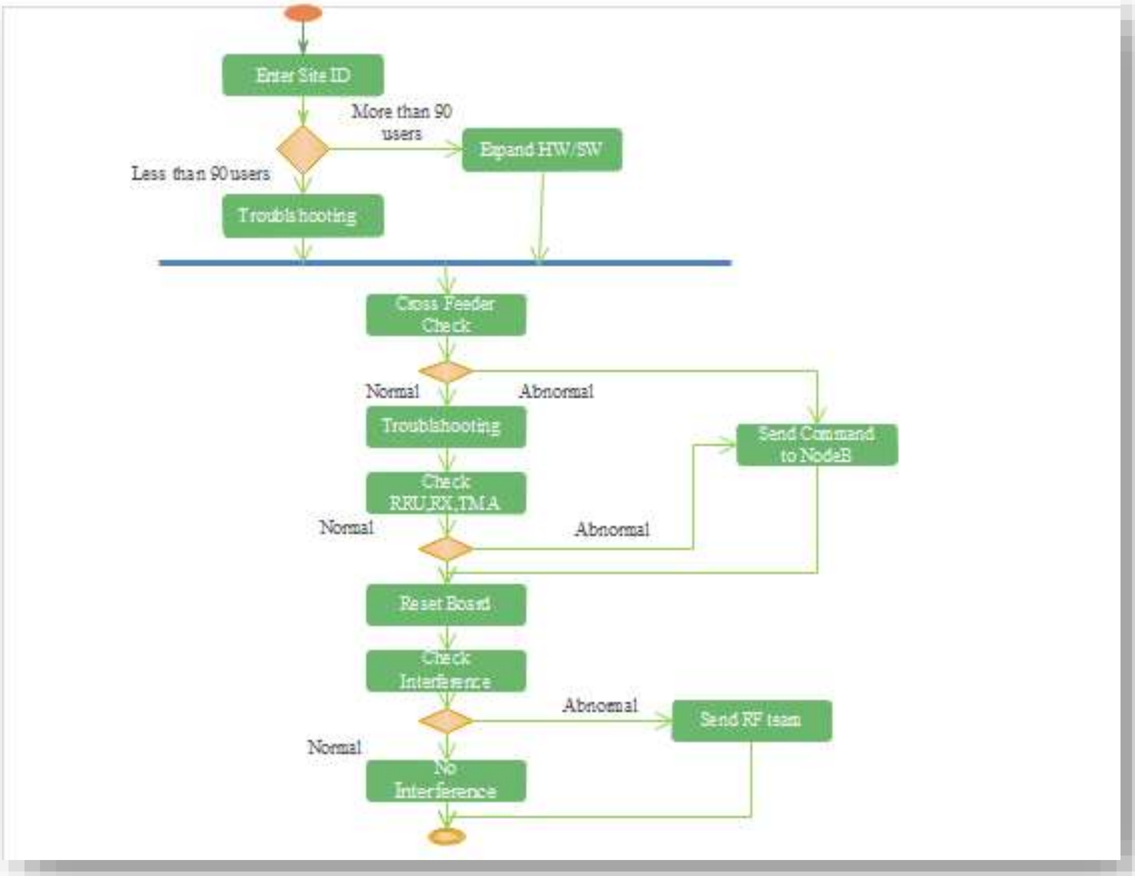


Figure 4.8 Activity diagram for tool in single-site mode

4.9 Activity Diagram for Multi-site Mode

Figure 4.9 shows the UML activity diagram that describes the sequence of actions between different objects when in multi-site mode.

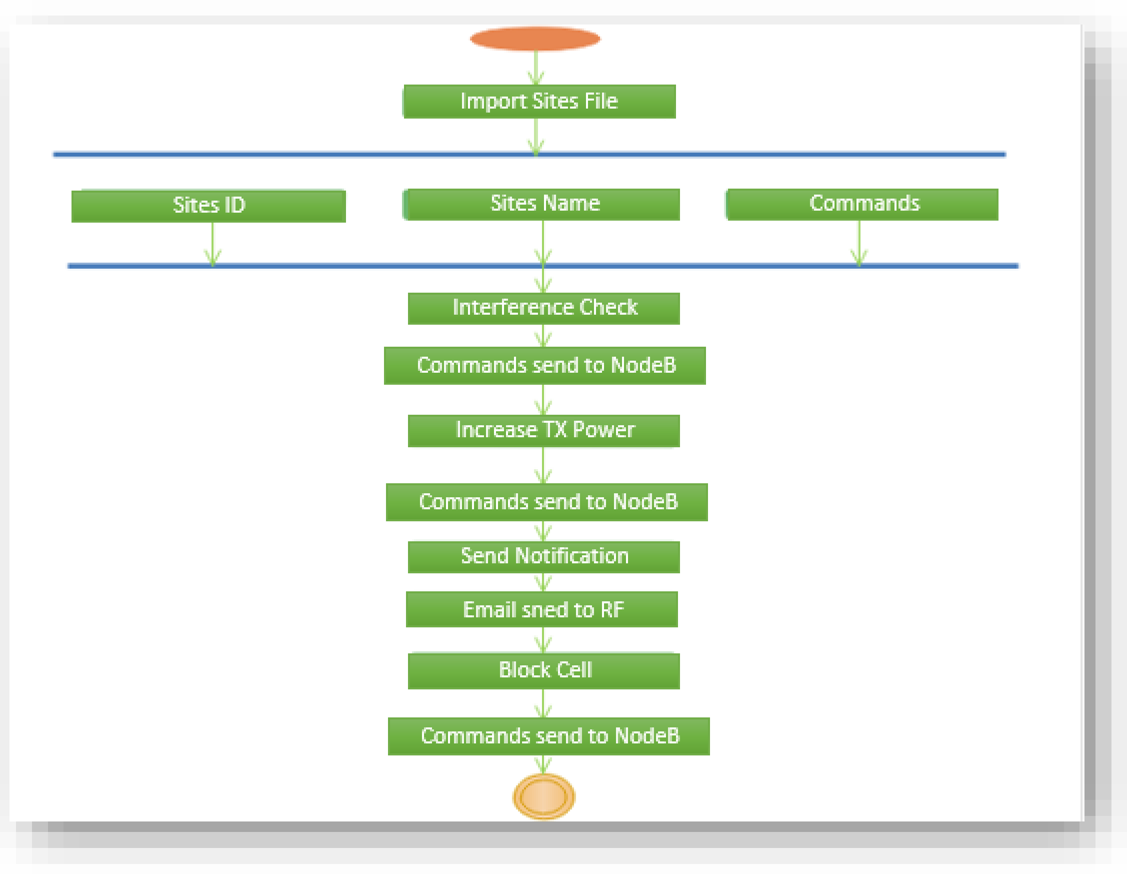


Figure 4.9 Activity diagram for the tool in multi-site mode

4.10 Deployment Diagram for Single-Site Mode

The UML deployment diagram expresses how the executables of the software application and the supporting resources are to be deployed on the hardware. Figure 4.10 shows the deployment of resources on the PC that is hosting the GUI tool and the relevant software, which is interfaced to and runs on NodeB. This diagram illustrates the relevant module deployments for single-site mode.

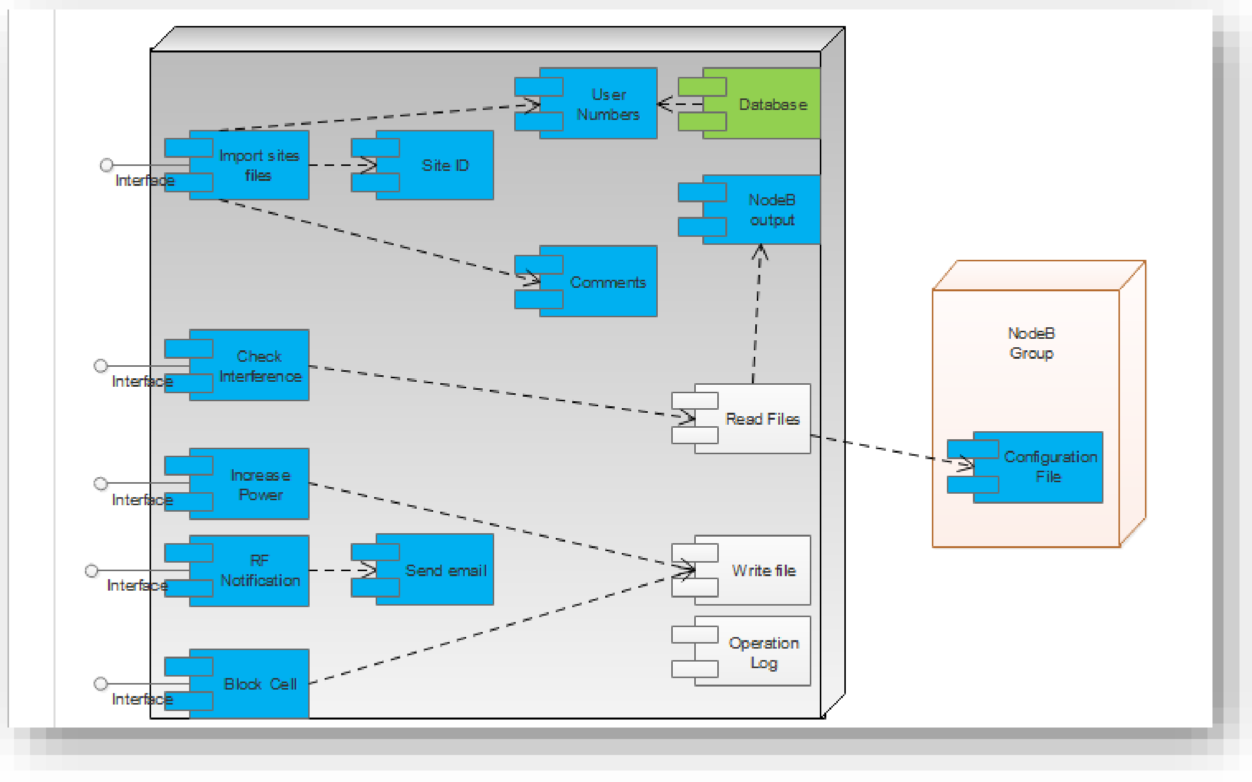


Figure 4.10 Deployment diagram for the tool in single-site mode

4.11 Deployment Diagram for Multi-site Mode

Figure 4.11 shows the UML deployment diagram for the tool in multi-site mode. This diagram illustrates the relevant module deployments, showing the resources and executables that are deployed on the PC running the GUI tool and on the NodeB platform.

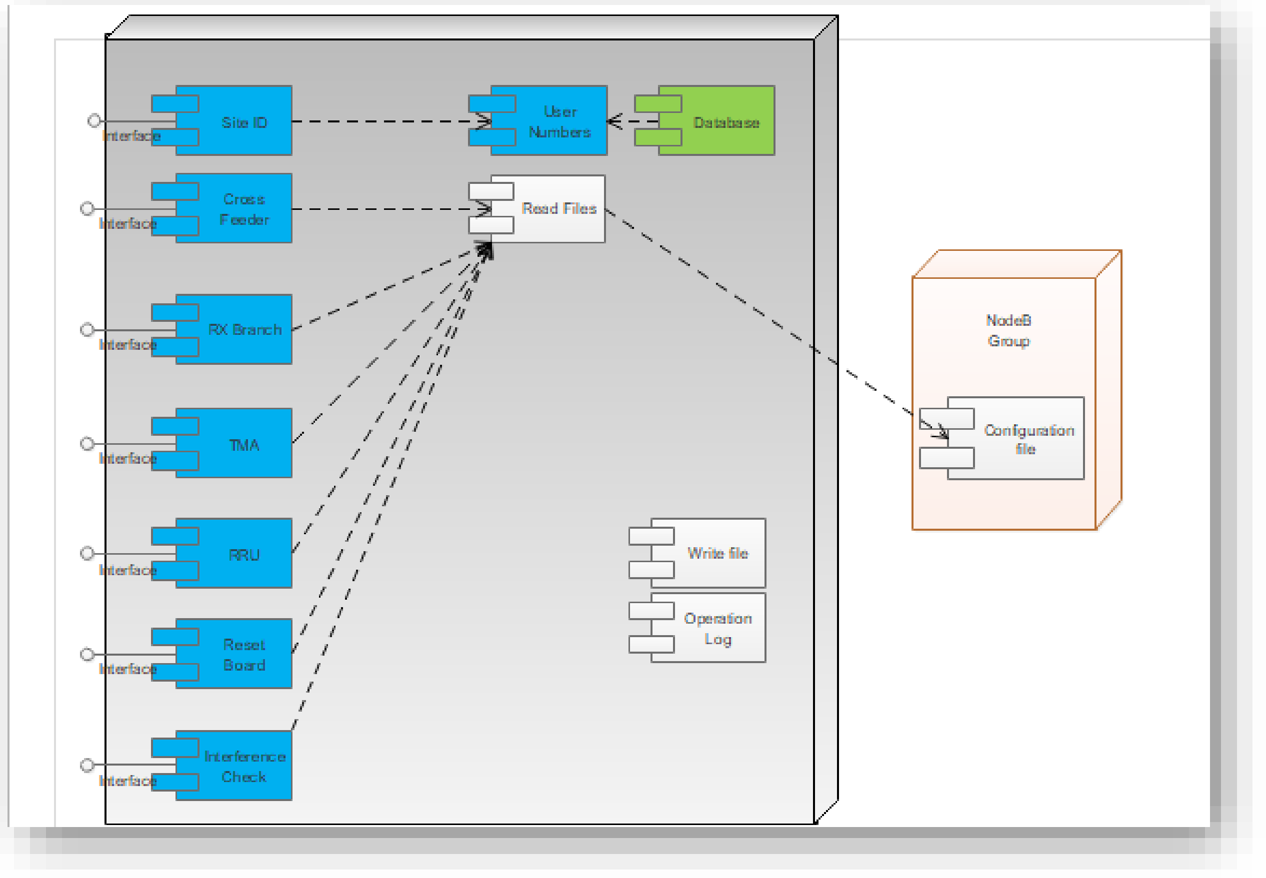


Figure 4.11 Deployment diagram for the tool in multi-site mode

4.12 Chapter Summary

This chapter introduced the principle of software modeling by giving a brief introduction to UML, which is an effective modelling method for any software system. It also discussed the classes and structures of UML that are relevant to this research, and which one suits our requirements. Furthermore, the actions that are performed by the user when using single-site mode and multi-site mode were introduced by means of sequence diagrams, and class diagrams were used to illustrate how the software was designed and to illustrate clear relationships between software objects.

CHAPTER 5: NODEB SYSTEM INTEGRATION AND CONFIGURATION

5.1 Introduction

This chapter discusses the Huawei NodeB 3900 interface and the physical structure, the board configurations and the role of each board in the NodeB. It also describes how it is commissioned through Universal Serial Bus (USB) and how the local maintenance terminal operates. In addition, it also covers the verification of services for the NodeB after the commission phase has been completed; lastly, it presents the tool used to accomplish this task and how this is handed over to the maintenance and operation team.

5.2 Configuration and Operational Details of Huawei NodeB 3900

The Huawei BTS 3900 (Base Transfer Station) is widely used by many mobile operators. It can be used as a Global System for Mobile (GSM) BTS, a Wideband Code Deviation Multiple Access (WCDMA) NodeB and a Long-Term Evolution (LTE) eNodeB. It has features such as high integration, low power consumption, and easy and quick deployment [9].

In terms of the logical structure of the NodeB, it generally consists of a Base Band Unit (BBU), and a Radio Unit, which could be a WCDMA Radio Frequency Unit (WRFU) or a Remote Radio Unit (RRU), depending on the design of the NodeB, The BBU3900[9]. Figure 5.1 shows the BBU connected to the WRFU or the RRU.

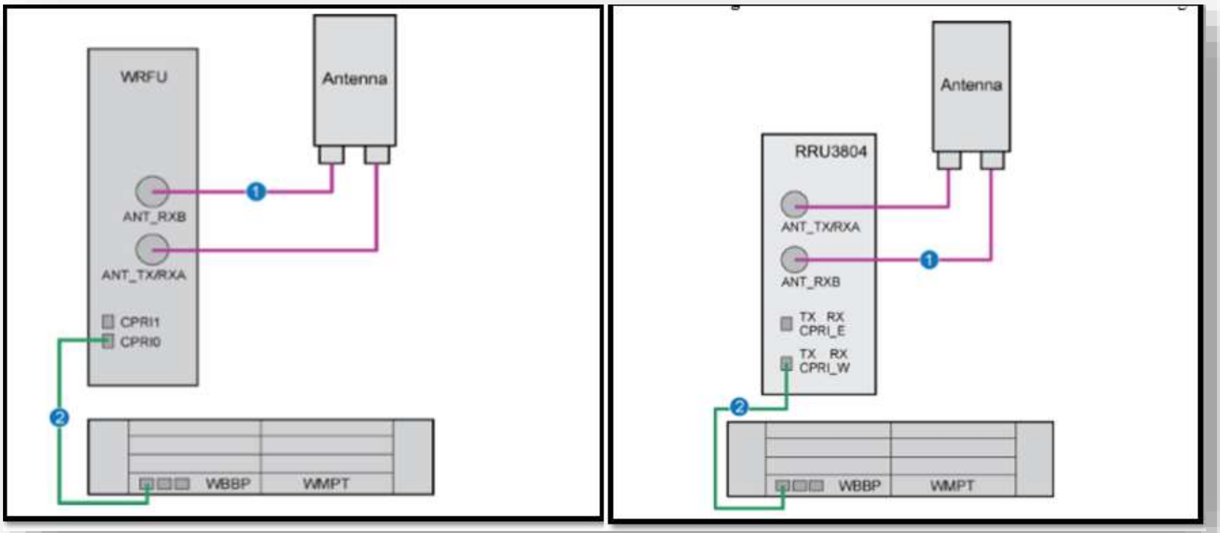


Figure 5.1 NodeB RRU and WRFU [9]

The logical structure of the BBU, an overview of which is given in Figure 5.2, consists of the following subsystems:

Control Subsystem: This is performed by the WCDMA Main Processing and Timing (WMPT) system. It provides operation and maintenance to NodeB, timing synchronization from an external clock, and processing of signals. Transmission resources connect to this board; if it is faulty, the NodeB will be completely isolated from the network and it will be out service [9].

Baseband Subsystem: This is implemented by the WCDMA Base Band Processing Unit (WBBP). It contains the Uplink (UL) and Downlink (DL) resources required by the NodeB to provide services to users [9].

Transport Subsystem: This is carried out by the WMPT and the Universal Transmission Possessing Unit (UTRP). Major tasks performed by the transport subsystem are:

- Containing the ports that provide transmission resources to NodeB; it can be Fast Ethernet (FE), Fibre or E1 port.
- Operation and maintenance channel between the NodeB, RNC and U2000 (U2000 is

the Node used to operate and maintain all Huawei equipment nodes remotely).

Power Module: This is responsible for converting +24 V DC or -48 DC power into power that is required to operate the boards and an external alarm, such as a smoke detector. Fire alarms are also connected to this module [9]. Figure 5.2 shows the logical structure of a BBU.

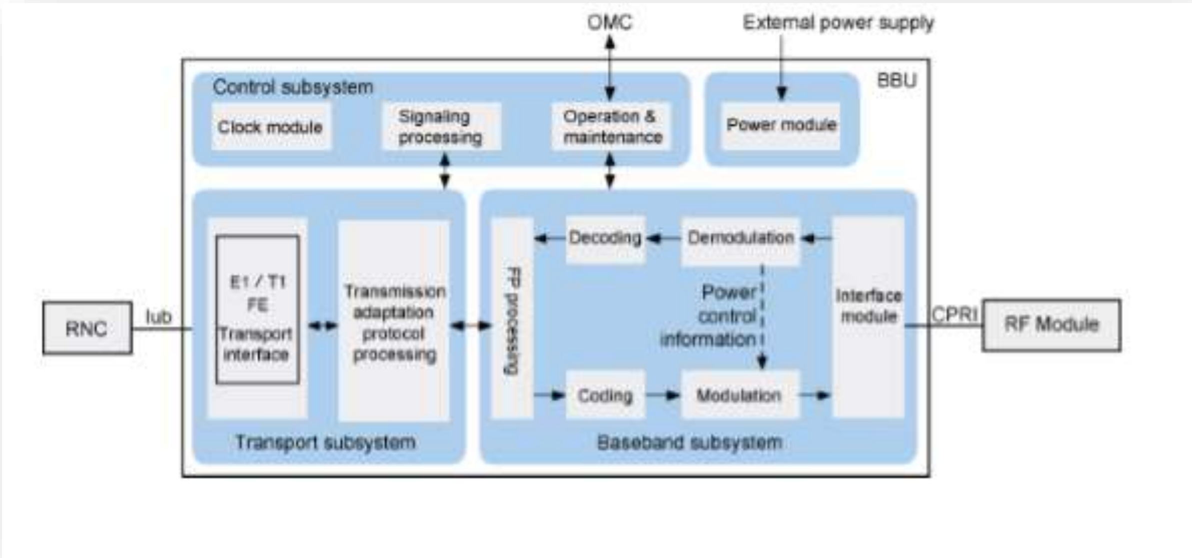


Figure 5.2 Logical structure of BBU [9]

5.3 An Overview of the Huawei NodeB 3900 Commissioning Procedure

The steps listed below summarize how the Huawei NodeB is commissioned and integrated into the Universal Terrestrial Radio Access Network (UTRAN). There are many ways to commission the NodeB: it can be done locally by USB (this is the easiest method), by means of local access through the Local Maintenance Terminal (LMT), and remotely from the Operation Centre. In this research, the first two methods are explained. The USB method will be discussed first [8]. To commission the NodeB with USB, the steps below are followed:

- NodeB is powered on and connected to the transmission equipment (IP over ETH or IP over fibre), depending on the transmission resources at the location [8].

- The configuration is executed at the RNC side Iub interface between RNC and NodeB [8].

- Cells are configured in the Mobile Switching Centre (MSC) to provide voice services

[8].

- The subcontractor team that is to commission the site needs to get three files from the Huawei RNC engineer's NodeB software, BootROM and the Data configuration file. Figure 5.3 shows the format of the files needed when commissioning the NodeB [8].

NodeB software: \\NodeB\Software\Software.cspBoot
BootROM: \\NodeB\Software\Firmware.csp
Data configuration file: \\NodeB\Configuration\NodeBCfg.xml

Figure 5.3 NodeB configuration files [8]

- NodeB software and BootROM can be downloaded from Huawei support, which is generally common for all NodeB. It contains the version of the NodeB and the software for running the board, but the data configuration file has to be created by the RNC engineer who needs to follow the steps in Figure 5.4 [8].

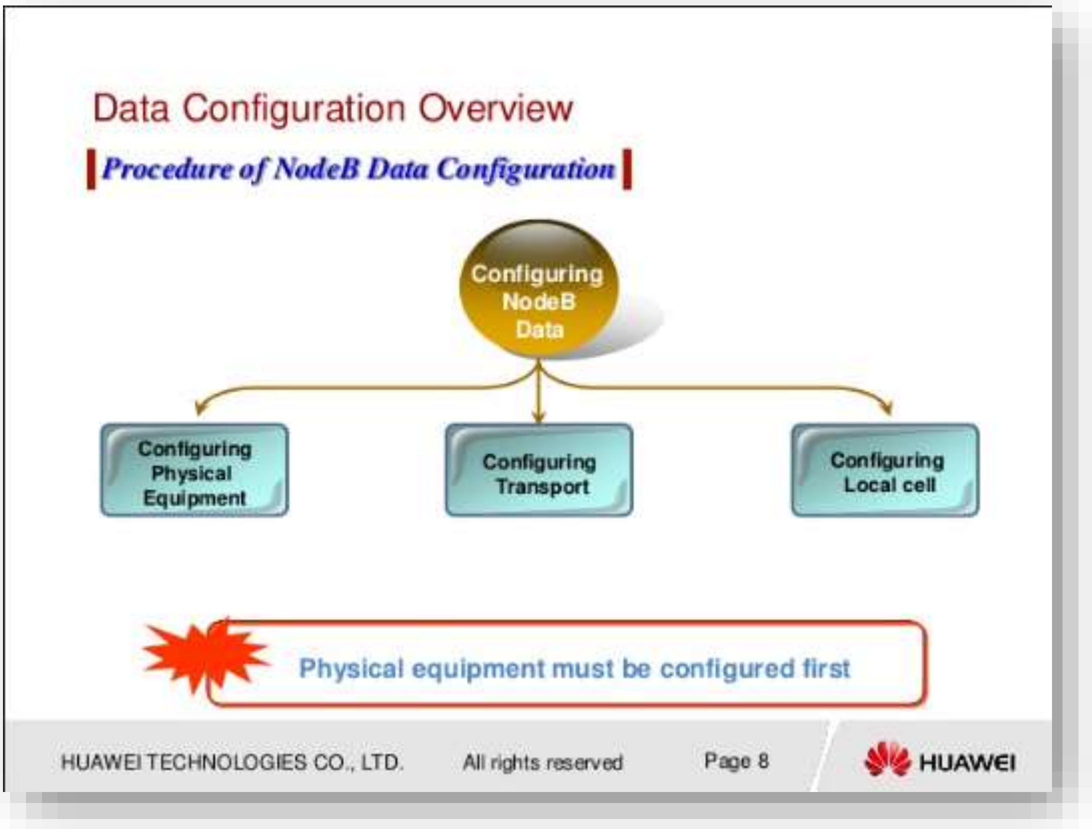


Figure 5.4 Steps of NodeB configuration [16]

- The RNC engineer needs to configure the physical layer first, which contains the NodeB boards, the site name, the site ID, and the physical layout of the NodeB. Thereafter, he needs to configure the transport layer, which contains the NodeB IP, the operation and maintenance IP, signaling and user plan parameters, the bandwidth supported by the NodeB and other transmission parameters that need to be negotiated with the transmission team. Thereafter, the RNC engineer needs to configure the cell or the radio part, which generally contains the cell ID, cell name, and maximum output power of the cell, the Location Area Code (LAC), and the Service Area Code (SAC) [8].

- Once the team enters the site for commission, they have to call the NOC to avoid any security alarms appearing at the site [8].

- After the hardware check has been completed, the site engineer needs to find the BBU serial number, as shown in Figure 5.5 [8].

Figure 5.5 Serial number for BBU [8]

5.4 USB Commissioning

Huawei NodeB 3900 has the capability to read data from USB. Thus, it is necessary to put the NodeB configuration, BootROM and software files on the USB and to insert these into the NodeB. The following steps explain how to use USB to commission Huawei NodeB [8].

- Insert the USB provided by Huawei into the USB port that is available in the BBU; if it is detected correctly by the NodeB, the following action takes place within a few seconds: the LED in the USB and WMPT board starts flashing ON and OFF for a certain period with a specific frequency, downloading the NodeB software file. Figure 5.6 shows the LED in the WMPT board [8].

Figure 5.6 LED in WMPT [8]

- The NodeB will start downloading the files from the USB, the LED on the USB and WMPT board will start flashing ON and OFF for a certain period of frequency to download the NodeB configuration file, as illustrated in Figure 5.6 [8].

- The NodeB software is activated once the download has been completed. Certain actions by the LED in the WMPT board will be observed, starting from flashing ON and OFF for a certain period to indicate that the NodeB software is being activated [38].

Figure 5.7 LED in USB when inserted in WMPT [8]

- The NodeB will restart itself; once all the files have been downloaded and activated successfully, certain actions will be performed by the LED until it stabilizes by displaying a green light, as shown in Figure 5.8 [8].

(1)

(2)

(3)

(4)

Figure 5.8 LED in WMPT and WBBP [8]

The entire NodeB commissioning process through USB can be summarized in Figure 5.9; this explains the status of the NodeB LED while it is reading the USB, and downloading the files from the USB to the NodeB, and while it is activating the relevant file in the NodeB, until the NodeB reaches normal running status.

Procedure	LED on the USB disk	RUN LED on the WMPT	During Time
Detecting USB disk	ON	Blinking (ON for 1s, OFF for 1s)	about 2sec
Downloading the software and the data configuration file from the USB disk	Blinking	Blinking (ON for 0.125s, OFF for 0.125s)	about 15min
Activating the software	ON	Blinking (ON for 0.125s, OFF for 0.125s)	about 10min
Resetting NodeB	ON->OFF->Blinking	ON->OFF->Blinking(ON for 0.125s, OFF for 0.125s)	about 5min
NodeB Running	Software Level Dependant	Blinking (ON for 1s, OFF for 1s)	–

Figure 5.9 LED status in USB and WMPT [8]

Thereafter, the site engineer will be able to access the NodeB and to start viewing the configuration version and any alarms, before enabling the services on the site. To finalize the commissioning phase, the site engineer needs to call the RNC engineer in the NOC to download the appropriate license in the NodeB and to unblock the cells to start testing for the NodeB services. Before starting the test, it is necessary for the site engineer to ensure that the cell orientation is correct and that there is no Crossed feeder interaction between the sectors. Figure 5.10 explains the details of the antenna orientation: sector one is called Alpha and is directed to tower north; sector two, which is called Beta, is on the right side to sector one in a clockwise direction; proceeding in a clockwise direction, one reaches sector three, which is called Gamma. Generally, the summation of the three sectors should be 360 degrees, but this is not mandatory. The azimuth of the antenna is decided by the Radio Planning team [8].

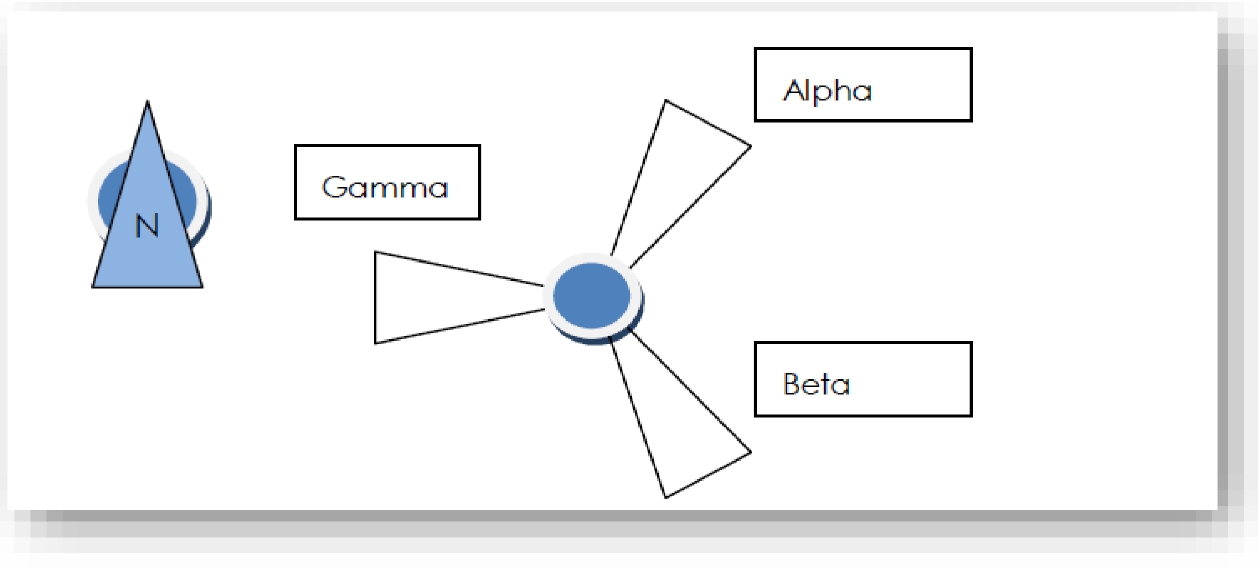


Figure 5.10 Antenna orientation [8]

After the antenna connection verification has been completed, it is necessary to start the single site verification process, which is process defined to verify the services in the new site, such as, for example, handoff, voice call, and internet access.

Single site verification: These processes are performed by the RF team to check the basic services and ensure that they meet the KPI requirements, for example, the download speed at the site, handover between 2G, 3G and 4G, and quality of voice communication; also, the radio parameters need to be audited during this process. Below is an example of the radio parameters that need be checked [27]:

- Verifying the coverage area by checking the configuration of the Receive Signal Code Power (RSCP) and the signal to noise ratio (Ec/Io) at the centre of cell and edge [27]
- Confirming that the Scrambling code is configured correctly, as planned [27]
- Applying the power setting for the NodeB cell [27]
- Configuring the neighbouring cells [27]
- Configuring 3G/2G selection and reselection: this is a very important parameter that is used to switch the phone from 3G to 2G when the signal of 3G services is weak [30].

This process is performed by the RF team, who should be equipped with the right tools that are necessary to accomplish this task [27]. Examples of tools used during single site verification are:

- A UMTS scanner, like Agilent E6474A [27]
- A UMTS user equipment (UE) test with a sim card [27]
- Global Positioning System (GPS) [27]
- YBT250 to identify sources of interference [27]
- Digital camera [27]
- Vehicle [27]

The final point of the USB Commissioning approach is to fill in the checklists that contain the hardware checks, the quality of the installation, the commission tests and the RF services tests. The results of all these tests need to be documented and kept in the site manual to be handed over to the operation and maintenance team.

5.5 Commissioning using LMT

In this method, the project engineer needs to obtain the Huawei local maintenance software and to install it on his laptop, so that he can commission the NodeB locally after installation and power on. The following points summarize this approach [8]:

- Connect the Laptop to the NodeB through the FE cable, as shown in Figure in 5.11

Figure 5.11 Laptop connected to NodeB via FE [8]

- Set the IP for the laptop on the same network segment as the NodeB maintenance IP. It is necessary to set the IP Address to 17.21.2.33 and the mask at 255.255.0.0 in the laptop, as shown in Figure 5.12 [8].

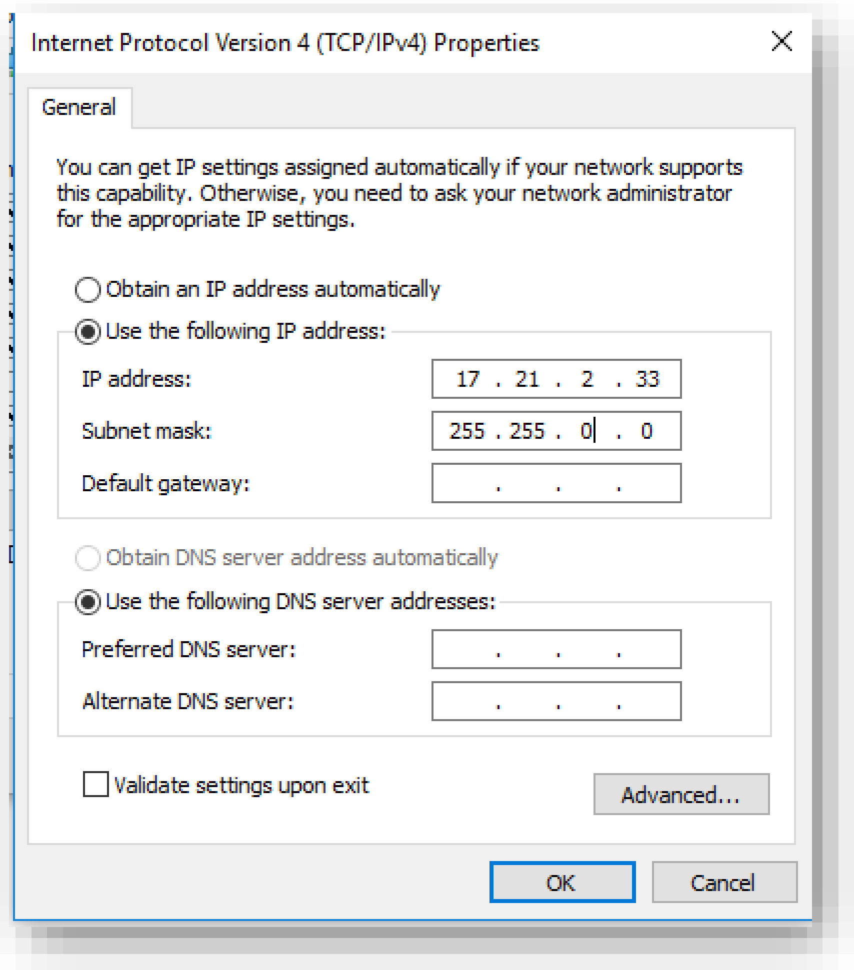
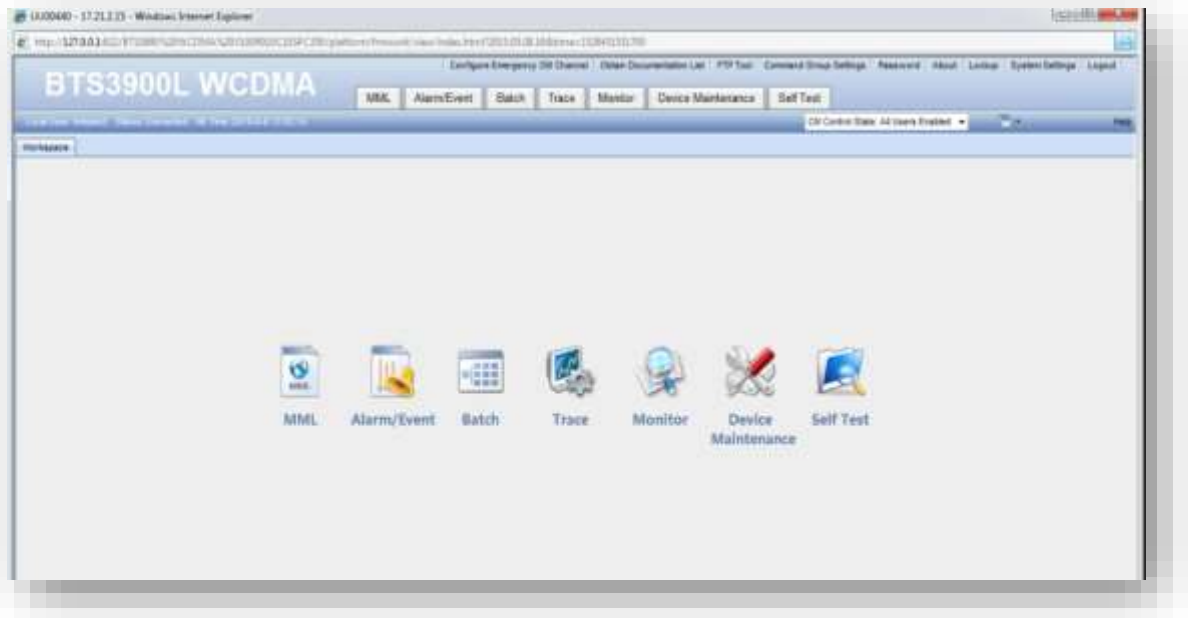


Figure 5.12 Setting NodeB and Laptop IP [8]

- Start using the local maintenance terminal by inputting the NodeB user name and password. Once logged-in successfully, the user should be able to browse the NodeB. Figure 5.13 shows these two steps.



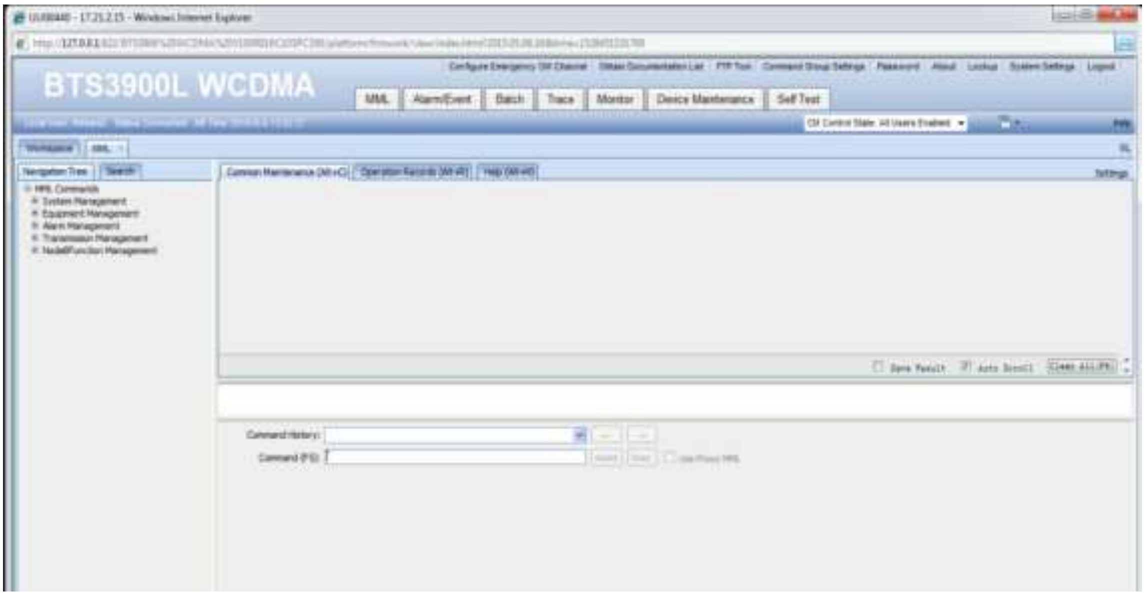


Figure 5.13 Login to NodeB [8]

- Upgrade the Boot ROM software first by uploading and activating the software on the NodeB. Figure 5.14 shows this step.

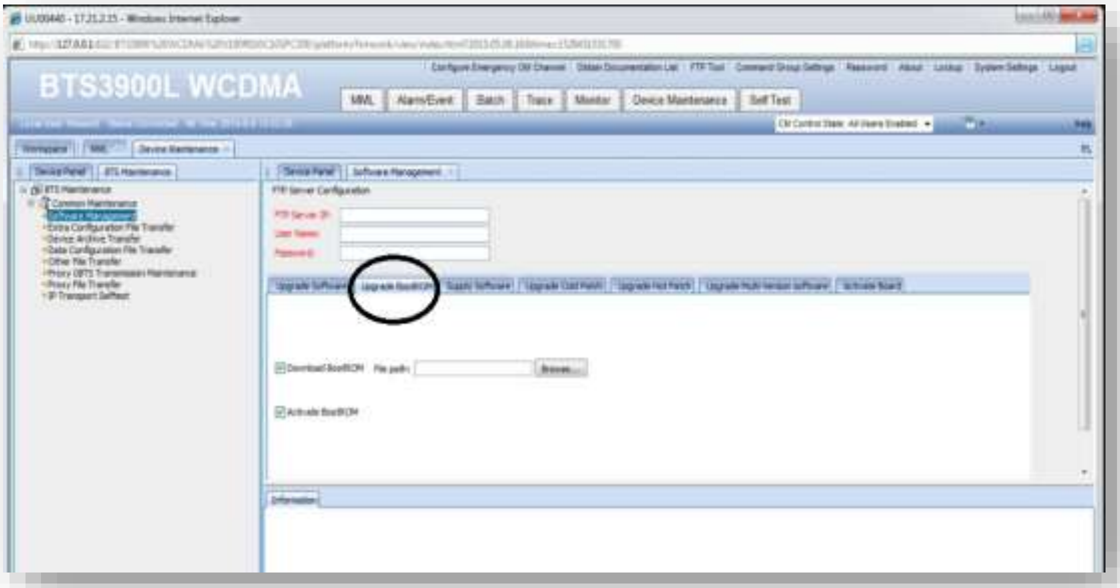


Figure 5.14 Boot ROM upgrade [8]

- Thereafter, upgrade the NodeB software by downloading it to the NodeB and activating it. Once this task has been completed, the NodeB restarts and will take around five minutes to run normally, as in Figure 5.15 [8].

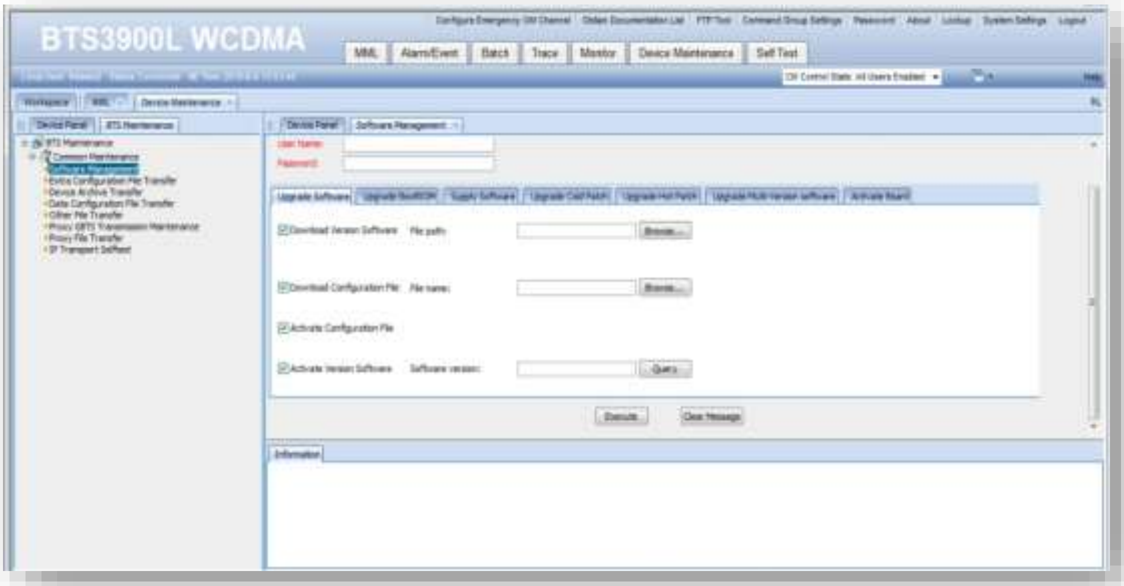


Figure 5.15 Activate NodeB software [8]

- Thereafter, upload the NodeB configuration file, which contains the NodeB configuration physical layer, transport layer and radio layer. Once the upload has been completed, activate the file and restart NodeB. It will take around five minutes to restart the services in the NodeB (as in Figure 5.16). After activation of the software and the configuration, NodeB should be in a normal run mode. Check the LED status to verify the run state of the NodeB, as in Figure 5.17 [8].

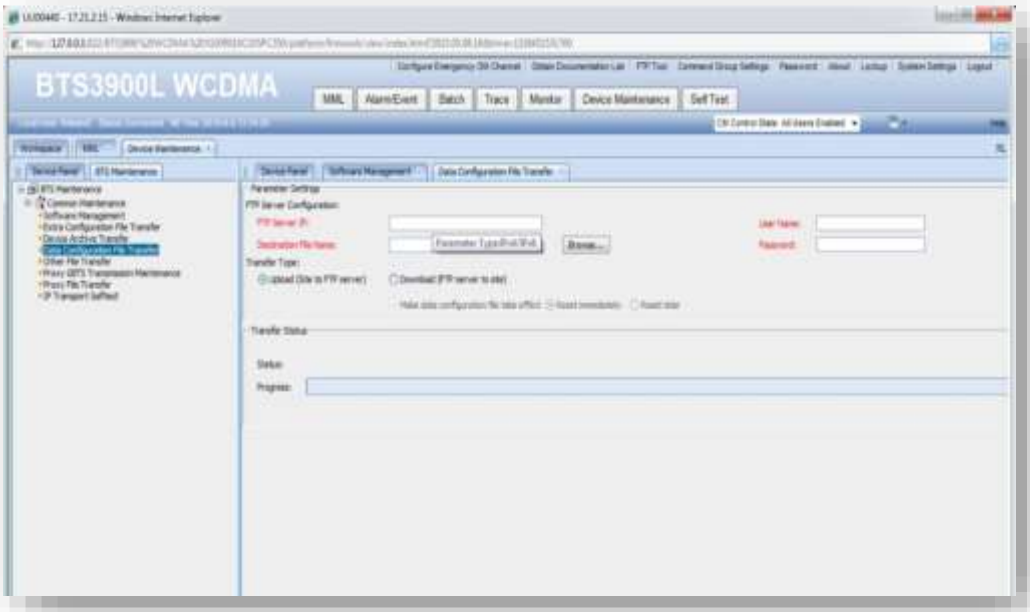


Figure 5.16 Activate NodeB configuration and login to NodeB [8]

When Huawei NodeB 3900 is running normally, the LED in WMPT board will be blinking ON for one second and OFF for one second. The WBBb LED board will follow the same pattern. All LEDs status for the boards in Huawei NodeB 3900 should be as described in Figure 5.17, which indicates normal operation.

LED	LED Status	Comments
WMPT Run LED	Blinking (ON for 1s, OFF for 1s)	
WMPT Alarm LED	OFF	Unless alarms are present
WMPT Act LED	ON	
WBBPb Run LED	Blinking (ON for 1s, OFF for 1s)	
WBBPb Alarm LED	OFF	Unless alarms are present
WBBPb Act	ON	
CPRI LED	ON	OFF if not configured, RED is RRU is not seen

Figure 5.17 NodeB running status after activation [8]

- Once the NodeB is back to its normal running status, the field engineer needs to inform the RNC engineer to upload the NodeB license from the U2000 server and then to unblock the cells to start checking for alarms and other related issues in the NodeB. Single Site Verification (SSV) can be performed to make the site ready for handoff to the maintenance team.

5.6 Chapter Summary

This chapter introduced the hardware and logical structure of the Huawei NodeB3900. The commissioning procedures by means of USB and LMT were reported. The chapter also discussed the requirements of integrating the Huawei NodeB with RNC and described how the services were tested and verified by different team. In addition, the tool used by the RF team during the SSV stage and how the site is transferred from the project team to the operation and maintenance team were discussed.

CHAPTER 6: IMPLEMENTATION

6.1 Introduction

In this chapter, a proof of concept (PoC) implementation is presented for a GUI tool to be used for RTWP troubleshooting in a 3G RAN. The tool will be used by the engineer in charge of operation and maintenance of the radio side of the NOC. The proposed solution is discussed in detail in this chapter, as well as how the GUI tool is used to help an operation engineer to fix the RTWP issue both in a single site and in multiple sites in a 3G network. Sometimes the RTWP alarm affects a cluster of sites for the following reasons:

- There is a higher than sustainable number of users in the cluster, for example, a high traffic event is being experienced at the site.
- Interferences at the site causes connections to be dropped or high error rates.

Testing for all the scenarios and functionalities of the designed solution is presented in this chapter.

6.2 RTWP Tool implemented in Single-site Mode

The GUI tool is designed to support the operation engineer in fixing the RTWP issue in a single site. To operate the configuration file effectively from the NodeB, some statistical information about the current user served by the NodeB is needed. In the designed tool, therefore, the configuration file is read from NodeB; this saved file is called the ‘Read File’. The statistical file that contains the number of users is generated by the tool. The file that contains the new command to be executed by NodeB is called the ‘Write File’, which is also generated by the tool. The commands that will be sent to NodeB for execution and further troubleshooting are thus contained in the Write File. Figure 6.1 shows the GUI for this tool, which has been developed by using the MATLAB GUI toolbox.

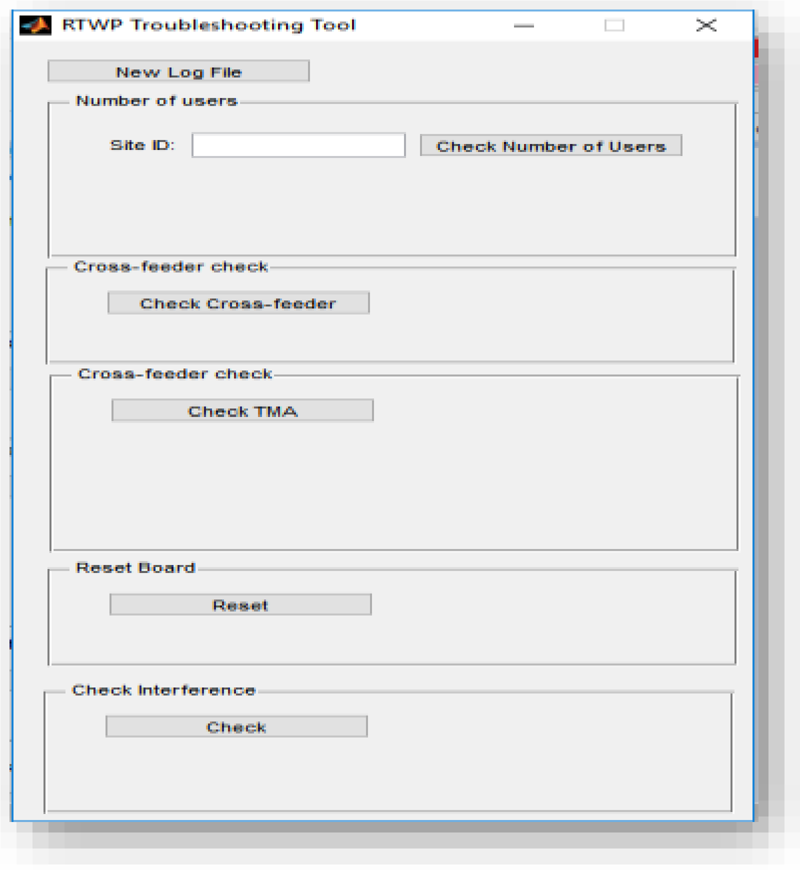


Figure 6.1 RTWP user interface for single-site mode

The user must enter the ‘site ID’ for the site that has a RTWP issue, as shown in Figure 6.2. The tool will search for this number in the database to find the corresponding user’s number served by the site at a specific time or at the time when the RTWP alarm appeared in the NOC. Table 6.1 shows the number of users on the relevant sites.

Table 6.1 Example of statistical file (Users’ numbers) copied from NodeB

site id	site name	number of users
1235	ASCF	90
1236	AWER	91
1237	QWER	92
1238	RTYU	93
1239	SERT	94
1240	DFGH	95

Figure 6.2 below shows the interface for the two sites where the user has entered the site ID for the sites to inspect. The interface view on the right shows a site, which has 91 users, thus one user above the specified threshold of 90 users. These displays show how the tool will react when the number of users is below or above the threshold.

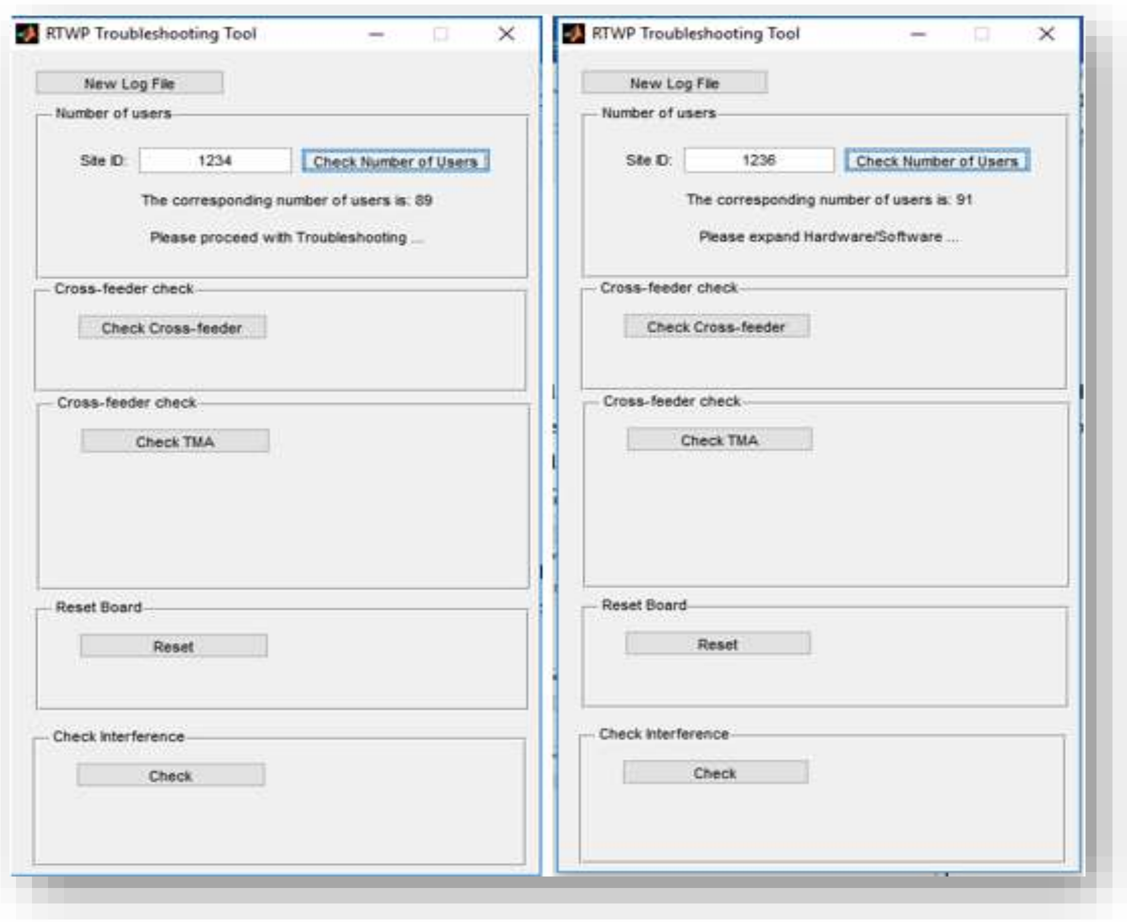


Figure 6.2 Number of users on the site

In the display below, the user has selected the crossed feeder check operation, which instructs the tool to get the Read File (viz. the NodeB configuration file) and to search for the expression “STR CROSFEEDTST”; this indicates that the crossed feeder check on NodeB has started. The output result is either normal or abnormal; Figure 6.3 shows both scenarios.

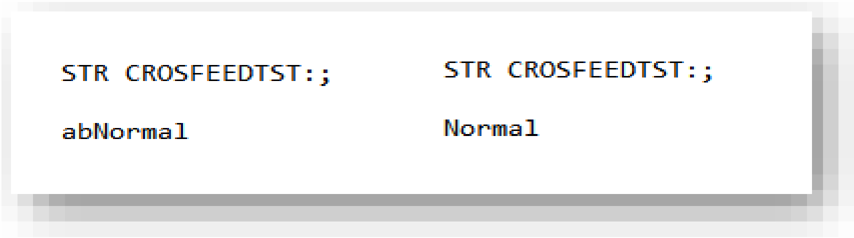


Figure 6.3 Crossed feeder status

Once the crossed feeder status has been determined, and if it is clear that there is a crossed feeder problem, the tool advises the user to send field a maintenance team to visit the site. If there is no crossed feeder issue, however, the tool asks the user to do further troubleshooting. Figure 6.4 shows the display for both of these scenarios.

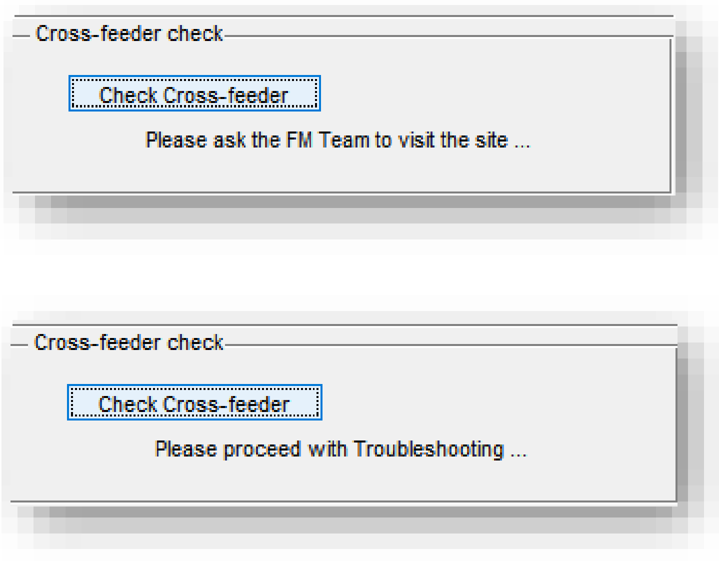


Figure 6.4 Output of crossed feeder check

The third step, once the user clicks on the TMA check, is to check the RX attenuation, the TMA configuration, and the RRU in the NodeB (in the dialog shown in Figure 6.2). The tool will go to the Read File and check the configuration, as illustrated in Figure 6.5; if anything abnormal is found in the configuration, the tool will generate a command to adjust the wrong parameter. The configurations checked by the tool are shown in Figure 6.5, Figure 6.6 and Figure 6.7.

```
%%/*128897098*/LST RXBRANCH:;%%
RETCODE = 0  Operation succeeded.

List RxBranch Configure Information
-----------------------------------
Cabinet No.  Subrack No.  Slot No.  RX Channel No.  Logical Switch of RX Channel  Attenuation(0

0            4            0         0               ON                            16
0            4            0         1               ON                            16
0            4            2         0               ON                            16
0            4            2         1               ON                            16
0            4            4         0               ON                            16
0            4            4         1               ON                            16

(Number of results = 6)
```

Figure 6.5 RX branch configuration

```
%%/*128902824*/LST RRU:;%%
RETCODE = 0  Operation succeeded.

List RRU/RFU Configure Information
----------------------------------
Cabinet No.  Subrack No.  Slot No.  Administrative State  RRU Topo Position  RRU Chain No.  RRU Position  RRU type  RF Unit Working Mode  Number of RX channels

0            4            0         Unblocked             TRUNK              0              0             WRFU      UMTS_ONLY             2
0            4            2         Unblocked             TRUNK              1              0             WRFU      UMTS_ONLY             2
0            4            4         Unblocked             TRUNK              2              0             WRFU      UMTS_ONLY             2
(Number of results = 6)
```

Figure 6.6 Remote Radio Unit (RRU) configuration

```
%%/*128906236*/DSP TMASUBUNIT:;%%
RETCODE = 0  Operation succeeded.

Display TMA Subunit Dynamic Information
---------------------------------------
Device No.  Device Name  Subunit No.  Online Status  Actual Work Mode  Actual Gain(0.25dB)  Actual Sector ID

3           3            1            AVAILABLE      NORMAL            48                   0
3           3            2            AVAILABLE      NORMAL            48                   0
4           4            1            AVAILABLE      NORMAL            48                   1
4           4            2            AVAILABLE      NORMAL            48                   1
5           5            1            AVAILABLE      NORMAL            48                   2
5           5            2            AVAILABLE      NORMAL            48                   2
```

Figure 6.7 TMA configuration

The scenarios when the configuration is correct, and when it is wrong, are discussed below. The first screenshot shows when the configuration is correct (Figure 6.8).

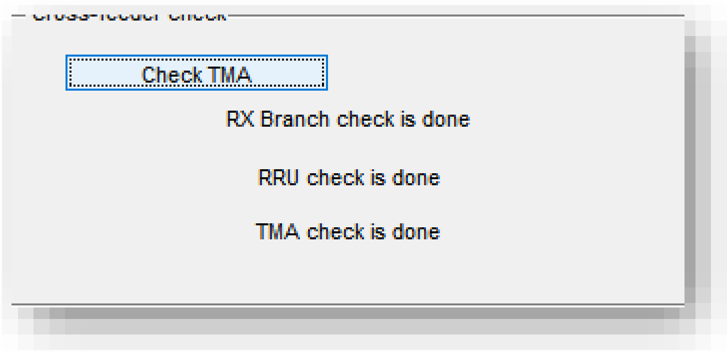


Figure 6.8 Output check for RX, TMA, and RRU

In the next display, a change was made to the configuration, where the attenuation was set to an invalid value. This was done to test that the tool corrects for this wrong parameter. The result below was obtained when the attenuation parameters were changed from 16 to 10. Figure 6.9 and Figure 6.10 illustrate this example.

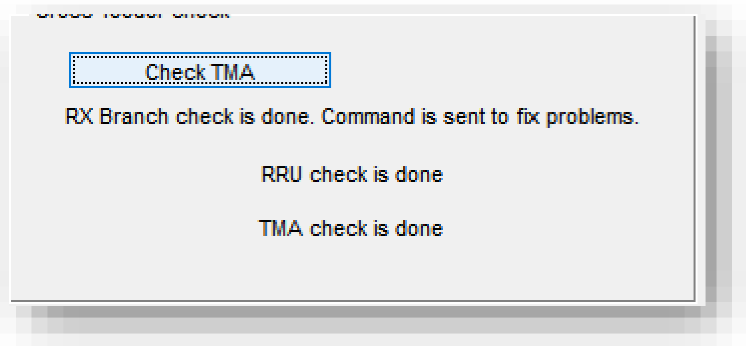


Figure 6.9 Message showing the wrong configuration for the RX branch

```
e Information
-------------
o.  Slot No.  RX Channel No.  Logical Switch of RX Channel  Attenuation(0.5dB)

    0         0               ON                            10
    0         1               ON                            16
    2         0               ON                            10
    2         1               ON                            16
    4         0               ON                            10
    4         1               ON                            16
```

Figure 6.10 Message showing the attenuation value set (incorrectly) to 16

These commands will modify the wrong configuration in the Write File, which should be executed in the Huawei NodeB to adjust the wrong parameter (Figure 6.11).

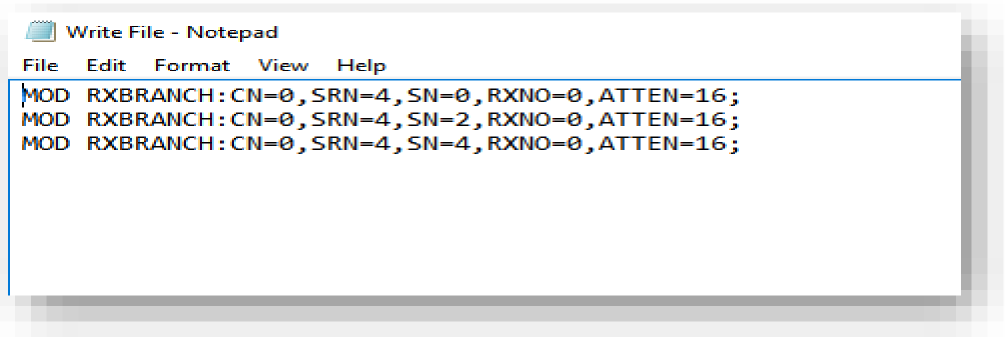


Figure 6.11 Command to correct the problems in the Write File

The fourth step is to send a reset command to the NodeB which causes a soft restart for the RRU board. Figure 6.12 shows that the reset was done successfully and the command sent to the NodeB. Figure 6.13 illustrate the board reset command in the Write File.

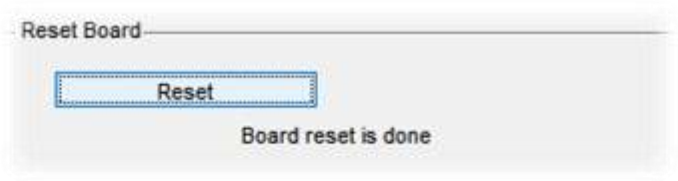


Figure 6.12 Board reset message

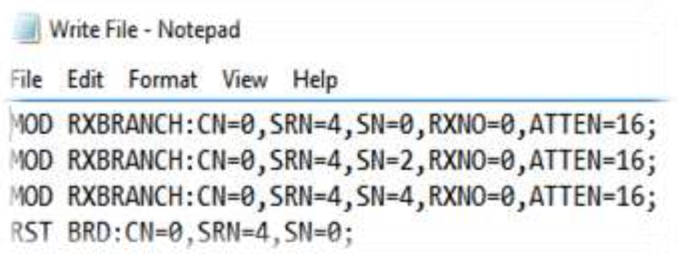


Figure 6.13 Board reset command in the Write File

The final step of the single-site mode of the tool is to check for interference. Once the user clicks on the command Check Interference, the tool will search the Read File for the “STR RFTEST” result, which shows whether there is interference. Figure 6.14 shows the output of the tool when there is interference, whereas Figure 6.15 shows the output when there is no interference.

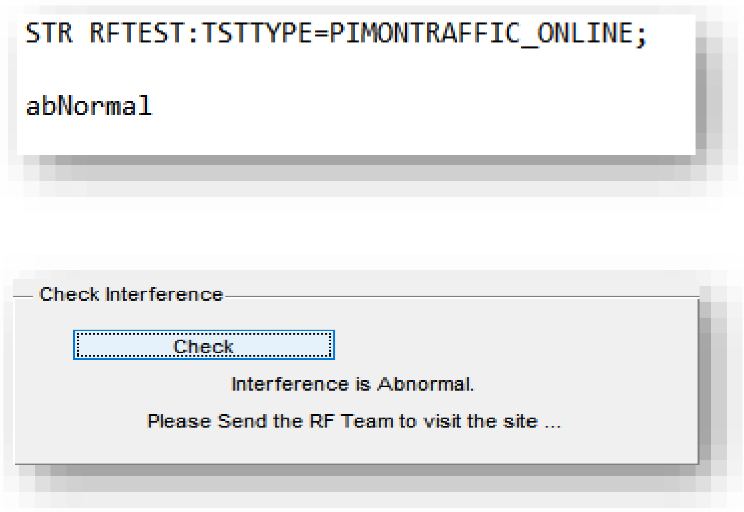


Figure 6.14 Output of interference checks when there is interference

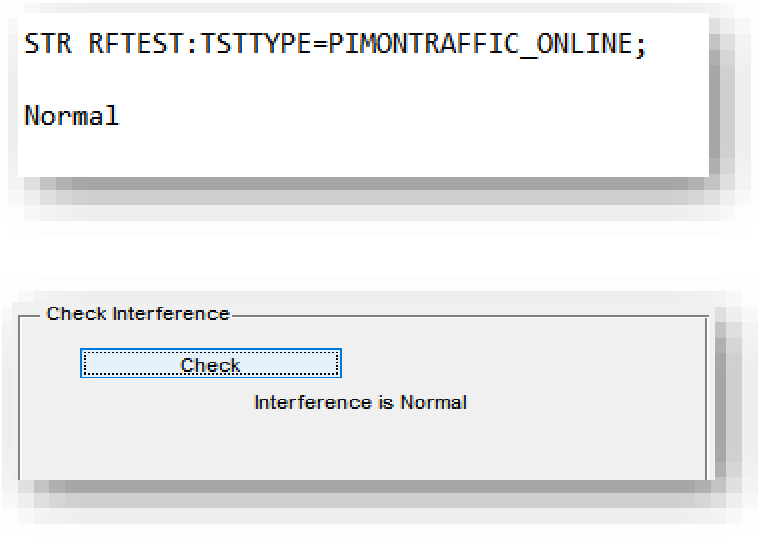


Figure 6.15 Output of interference checks when there is no interference

Finally, the operation log is designed to show all the operations performed in the GUI tool, as illustrated in Figure 6.16.

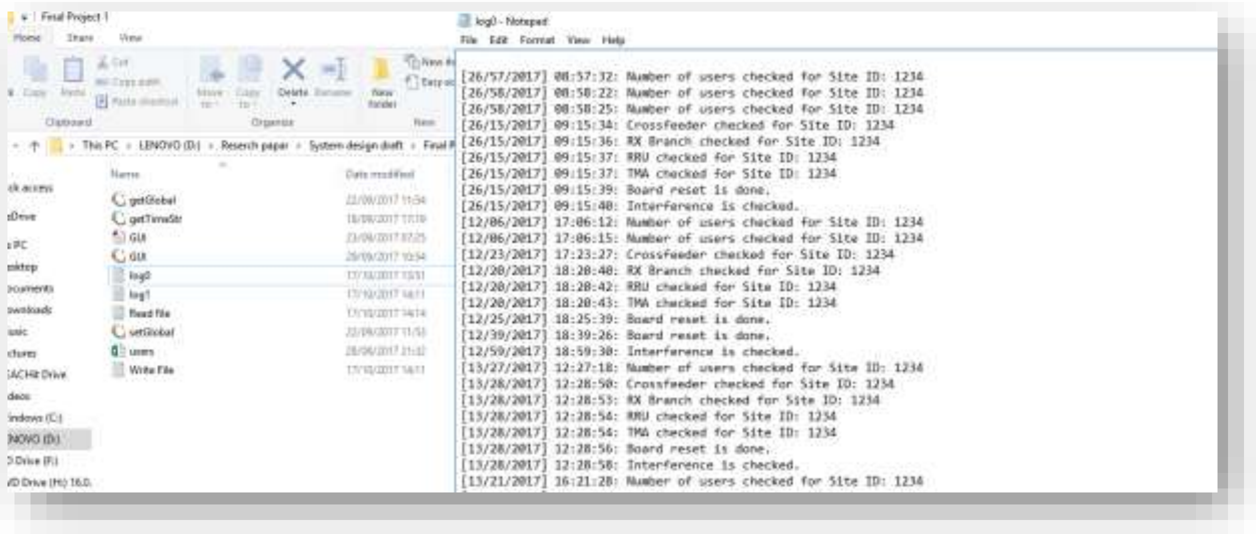


Figure 6.16 Operation log for correcting attenuation errors

6.3 RTWP tool implemented in Multi-site Mode

The second tool is used when RTWP alarms occur in multiple sites on the network. The first mode of the tool deals with single-site configurations (discussed above), while the second mode of the tool deals with multiple sites. This is illustrated in Figure 6.17, where 10 sites are affected by RTWP alarms.

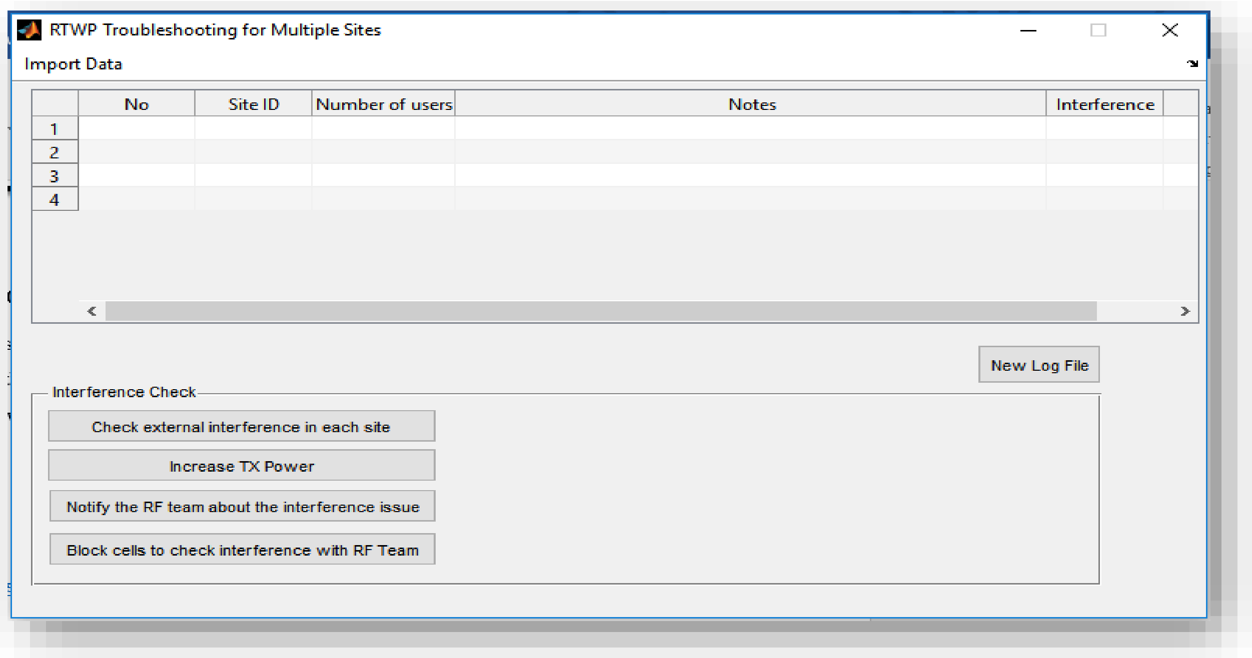


Figure 6.17 RTWP alarms for multiple sites

In this case, too, the tool needs to extract the user's number from statistics counters, which contains the current users' numbers on the affected sites. Table 6.2 shows the number of users along with the site ID.

Table 6.2 RTWP alarms for multiple sites

No	Site ID	Number of Users	Cell ID		
			Sector A	Sector B	Sector C
1	1122	69	11225	11226	11227
2	1121	70	11215	11216	11217
3	1234	71	12345	12346	12347
4	1345	72	13455	13456	13457
5	2132	0	21315	21316	21317
6	2333	74	23335	23336	23337
7	3213	99	32135	32136	32137
8	4111	76	41115	41116	41117
9	4321	91	43215	43216	43217
10	6111	78	61115	61116	61117

The user must import the configuration files of the sites that have an RTWP issue in the network by using the Import button (Figure 6.18).

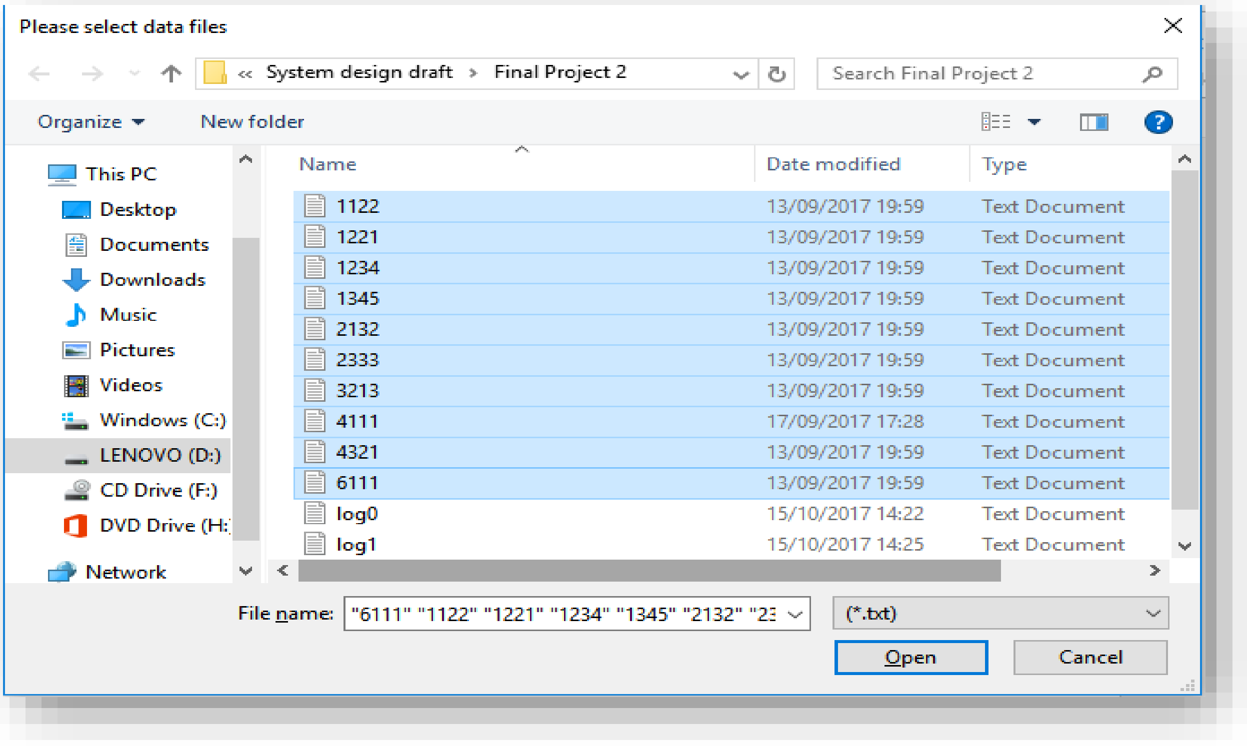


Figure 6.18 Importing multiple sites

Once the user has imported the configuration files, the tool will present the sites' identification numbers and the current number of the users served by the sites. Furthermore, it indicates whether the number of users on the sites is exceeding the capacity of the NodeB. The threshold in the design is 90 users per second. Figure 6.19 illustrates this operation.

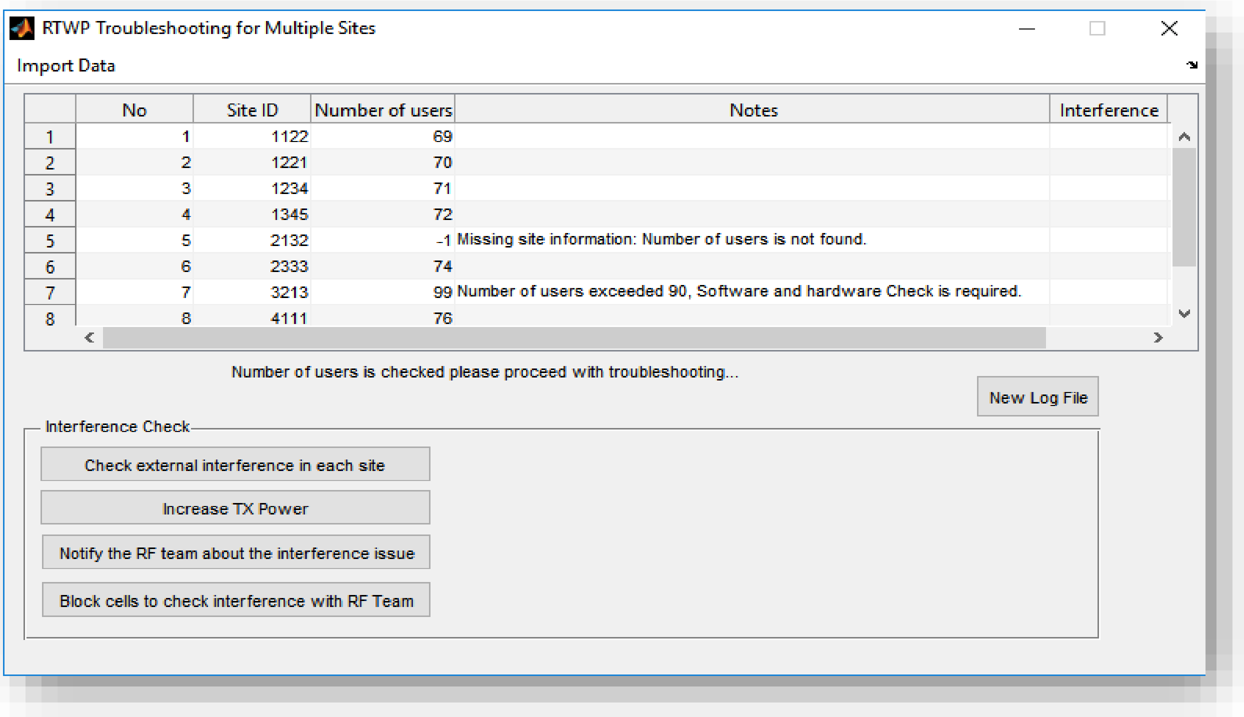


Figure 6.19 Checking the files for multiple sites

Once the user has clicked on the command to Check External Interference, the tool will check the Read File to see whether there is interference by searching for the "STR RFTEST" string. An example of one site (site #1), which does not suffer from interference is shown in Figure 20 – i.e. it shows NORMAL for site #1. Figure 21 shows the situation where site #1 also has interference.

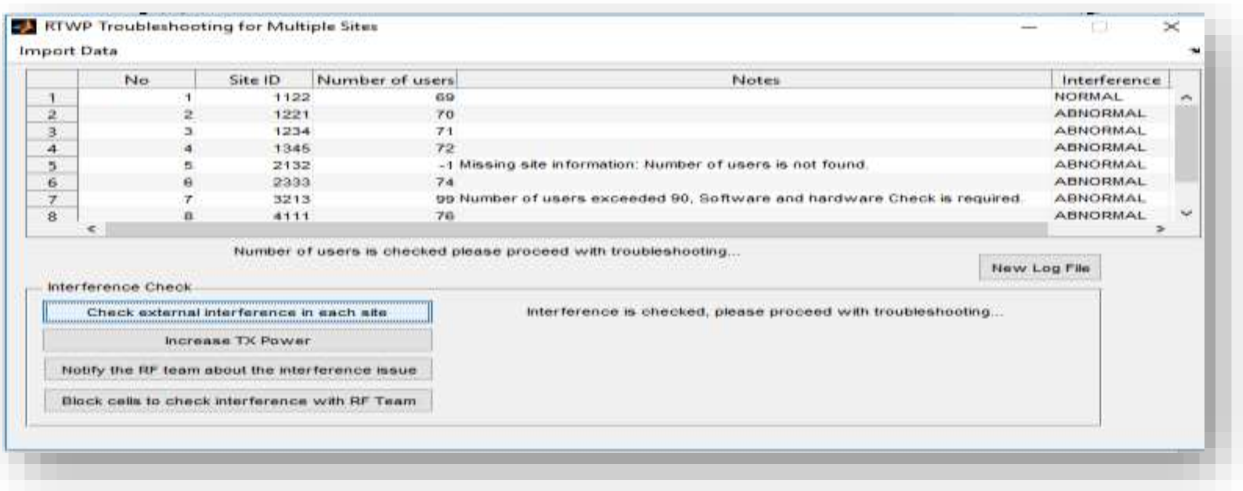


Figure 6.20 Result of interference check

If one or more of the sites have an interference issue, as shown in Figure 6.22, the user is asked to proceed with troubleshooting.

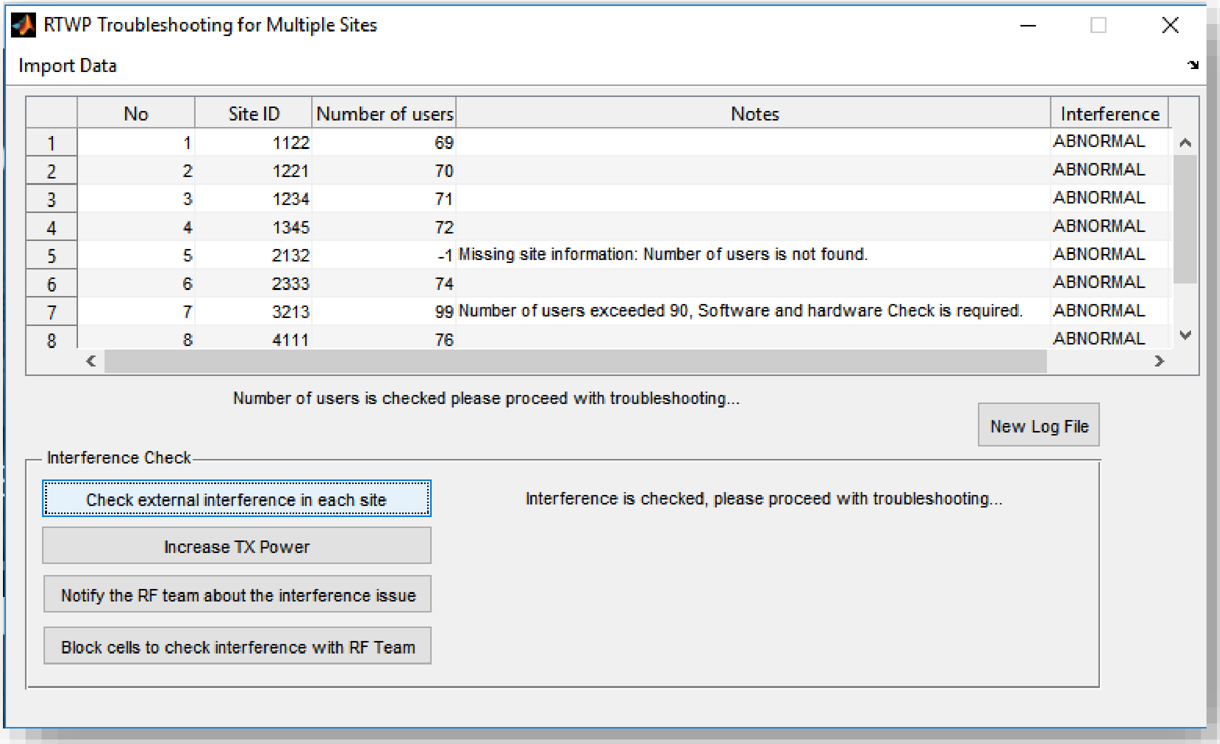


Figure 6.21 Troubleshooting the interference results

In the third step, the user will increase the TX power of the cell; this command will be written in the Write File, to send to the Huawei NodeB (Figure 6.22).

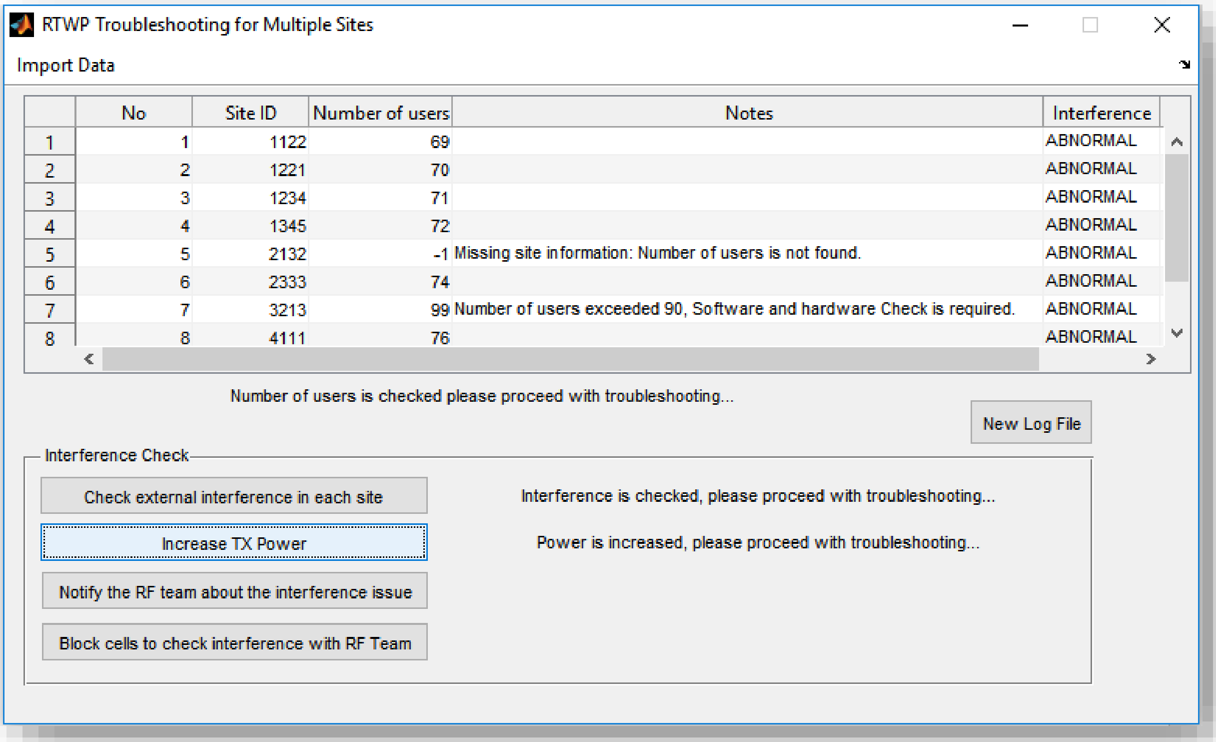


Figure 6.22 Message to increase power

The Write File will attempt to resolve the interference problem, as generated by the tool, as shown in Figure 6.23. The command in this file will be executed in sequence on NodeB.

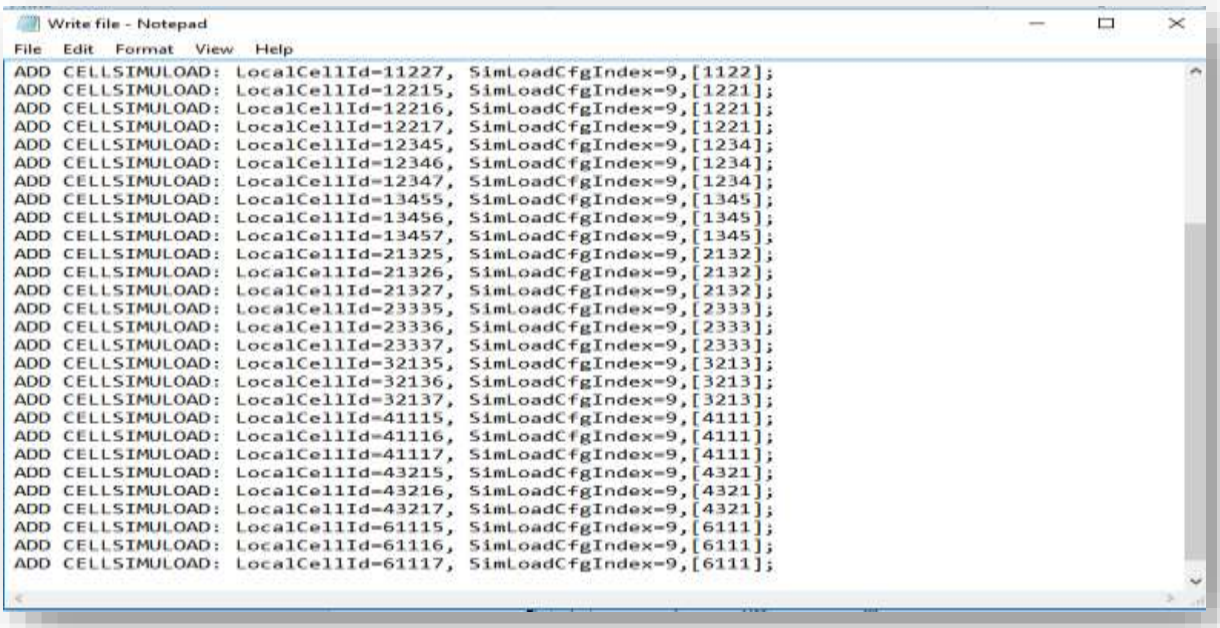


Figure 6.23 Power increase commands generated and stored in the Write File

The next step is to notify the RF team about the interference status on the relevant sites. Once the user clicks on Notify RF Team about the interference issue, the tool sends an automatic email and asks the maintenance team (i.e. the user of the GUI tool) to proceed with troubleshooting. Figure 6.24 and Figure 6.25 explain this step.

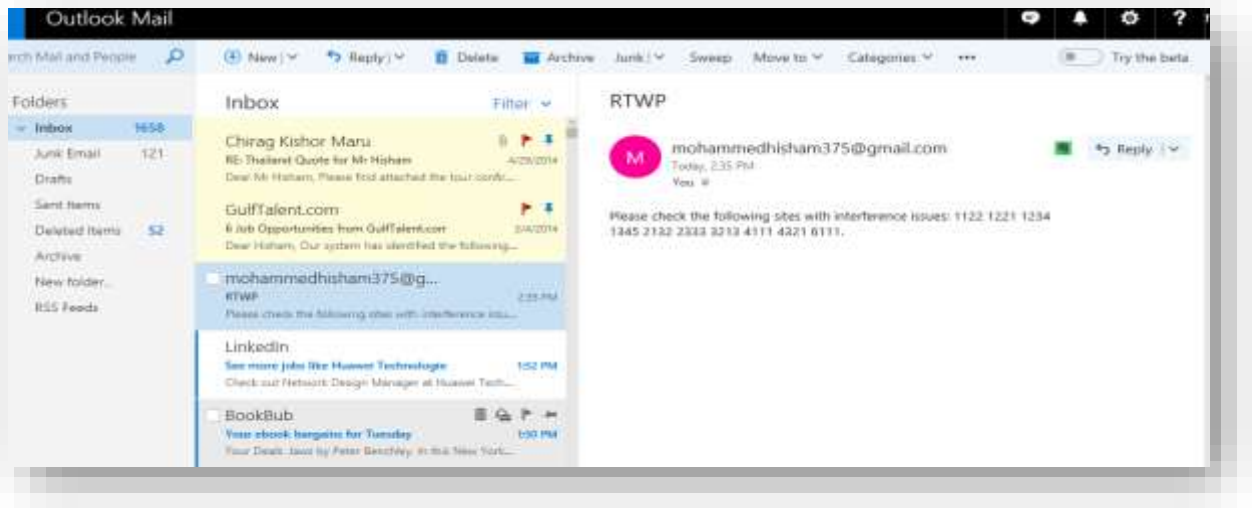


Figure 6.24 Email notification sent to RF team

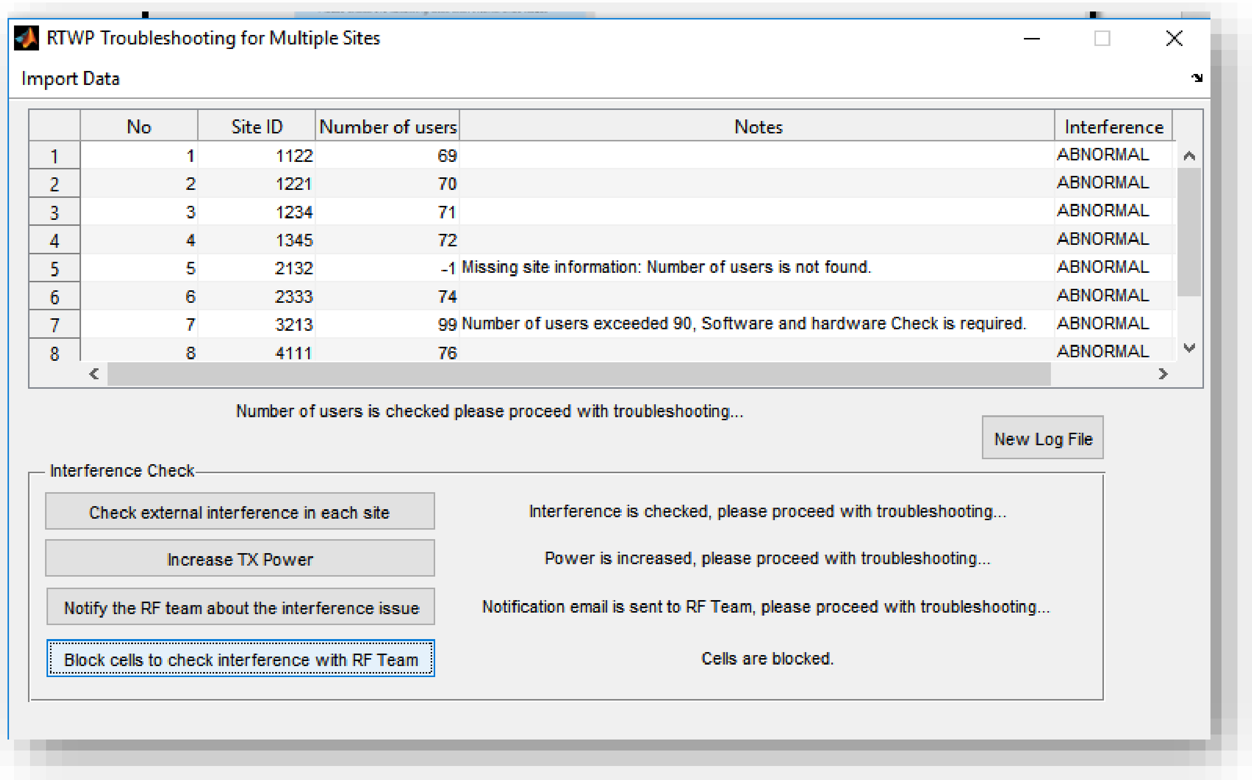


Figure 6.25 Block command sent

In the final step, the user needs to click the Block Cell button in the tool. The tool then generates block commands to block the services in the specific cells, so that the RF team can identify the sources of interference. Figure 6.26 shows the block command that can be executed in the NodeB to block the services.

```
Write File - Notepad
File  Edit  Format  View  Help
ADD CELLSIMULOAD: LocalCellId=43217, SimLoadCfgIndex=9,[4321];
ADD CELLSIMULOAD: LocalCellId=61115, SimLoadCfgIndex=9,[6111];
ADD CELLSIMULOAD: LocalCellId=61116, SimLoadCfgIndex=9,[6111];
ADD CELLSIMULOAD: LocalCellId=61117, SimLoadCfgIndex=9,[6111];
BLK CELL: CellID=11225, [1122];
BLK CELL: CellID=11226, [1122];
BLK CELL: CellID=11227, [1122];
BLK CELL: CellID=12215, [1221];
BLK CELL: CellID=12216, [1221];
BLK CELL: CellID=12217, [1221];
BLK CELL: CellID=12345, [1234];
BLK CELL: CellID=12346, [1234];
BLK CELL: CellID=12347, [1234];
BLK CELL: CellID=13455, [1345];
BLK CELL: CellID=13456, [1345];
BLK CELL: CellID=13457, [1345];
BLK CELL: CellID=21325, [2132];
BLK CELL: CellID=21326, [2132];
BLK CELL: CellID=21327, [2132];
BLK CELL: CellID=23335, [2333];
BLK CELL: CellID=23336, [2333];
BLK CELL: CellID=23337, [2333];
BLK CELL: CellID=32135, [3213];
BLK CELL: CellID=32136, [3213];
BLK CELL: CellID=32137, [3213];
BLK CELL: CellID=41115, [4111];
BLK CELL: CellID=41116, [4111];
BLK CELL: CellID=41117, [4111];
```

Figure 6.26 Block command generated and stored in the Write File

The tool also has operation logs that show any tasks performed by the user (Figure 6.27).

```
File  Edit  Format  View  Help
[17/09/2017] 16:09:16: Site ID(s): 1122, 1221, 1234, 1345, 2132, 2333, 3213, 4111, 4321, 6111 has/have been imported for troubleshooting.
[17/09/2017] 16:09:19: Interference is checked for Site ID(s) 1122, 1221, 1234, 1345, 2132, 2333, 3213, 4111, 4321, 6111.
[17/27/2017] 16:27:36: Power is increased for Site ID(s): 1122, 1221, 1234, 1345, 2132, 2333, 3213, 4111, 4321, 6111.
[17/35/2017] 16:35:29: Notification email is sent to RF Team regarding Site ID(s): 1122, 1221, 1234, 1345, 2132, 2333, 3213, 4111, 4321, 6111.
[17/37/2017] 16:37:47: Cells are blocked for Site ID(s): 1122, 1221, 1234, 1345, 2132, 2333, 3213, 4111, 4321, 6111.
[17/10/2017] 18:55:48: Site ID(s): 1345 has/have been imported for troubleshooting.
[17/10/2017] 18:55:54: Site ID(s): 1122, 1221 has/have been imported for troubleshooting.
[17/10/2017] 18:55:55: Interference is checked for Site ID(s) 1122, 1221.
[17/10/2017] 18:55:56: Power is increased for Site ID(s): 1122, 1221.
[17/10/2017] 18:56:07: Notification email is sent to RF Team regarding Site ID(s): 1122, 1221.
[17/10/2017] 18:56:07: Cells are blocked for Site ID(s): 1122, 1221.
[17/10/2017] 18:56:44: Site ID(s): 1122, 1221, 1234, 1345, 2132, 2333, 3213, 4111, 4321, 6111 has/have been imported for troubleshooting.
[17/10/2017] 18:56:46: Interference is checked for Site ID(s) 1122, 1221, 1234, 1345, 2132, 2333, 3213, 4111, 4321, 6111.
[17/10/2017] 18:56:48: Power is increased for Site ID(s): 1122, 1221, 1234, 1345, 2132, 2333, 3213, 4111, 4321, 6111.
[17/10/2017] 18:56:53: Notification email is sent to RF Team regarding Site ID(s): 1122, 1221, 1234, 1345, 2132, 2333, 3213, 4111, 4321, 6111.
[17/10/2017] 18:56:53: Cells are blocked for Site ID(s): 1122, 1221, 1234, 1345, 2132, 2333, 3213, 4111, 4321, 6111.
```

Figure 6.27 Operation log

6.4 Chapter Summary

This chapter presents the various tests and experiments that were carried out as proof of concept for the proposed tool, which is envisaged to support the operations engineer in solving RTWP issues in 3G RAN. An analysis was done to evaluate the effectiveness of the tool for both single-site and multi-site issues. This chapter discusses the various aspects of the proposed solution, such as step-by-step implementations of the tool, the results of each of these steps, and how to use the tool to resolve RTWP issues in both single-site and multi-site modes. The commission procedures used by the operations engineer and how the results are obtained when there is interference affecting a site.

CHAPTER 7: TESTING AND RESULTS

7.1 Introduction

This chapter discusses the experiment of testing the GUI tool to resolve a RTWP issue in various sites of the Oman Telecommunication Company's 3G network. According to the operation and regulations rules of the network, the engineer is not allowed to test any tool in a live network, and thus the GUI tool that was designed in this project will not be connected to a device running in the live network, as this may cause dropped connections and lead to potential expenses for the network provider. The engineer will obtain accurate data from the network from real fault situations, what will be replayed as simulations and imported into the tool. All the scenarios that the tool used in diagnosis and troubleshooting will be based on data that will be imported by the RAN back office engineer. The time it takes to resolve the issues depends on the skill of the engineer responsible for resolving the issues in the NOC. The senior engineers all know how to break down the problem by following a sequence of steps, starting with checking the configuration, then checking the traffic statistics, and finally reporting the feedback about the solution and closing the issue. The GUI tool helps with this process. The tool is user-friendly and designed to smooth operations with these maintenance tasks.

Five scenarios are tested in this chapter. The first scenario looks at a situation where a high number of RTWP issues occurs in a single site, caused by a wrong RX attention setting; the tool identifies the wrong parameter setting and then adjusts it to resolve the problem. In the second scenario, a high number of RTWP faults is found on two sites simultaneously; the RAN back office engineer uses the tool to identify the root cause of interference at both sites, and is required (as per the maintenance procedures) to highlight this issue to another department for further analysis and fixing. In the third scenario, the tool is used to solve an interference issue in two sites by generating the commands and executing them in the Huawei NodeB 3900. In the fourth scenario, the tool is used to resolve interference issues in one site after finding a wrong configuration in one of the parameters. Some scenarios cannot be tested, such as the crossed feeder issue and the restart board in NodeB. The final scenario tested looked at hardware or software expansion in NodeB to increase capacity. Finally, a comparison is presented between resolving a RTWP issue in the network with the tool and without the tool, to evaluate the benefits of using the GUI tool.

7.2 First Scenario

The RTWP KPI degradation was reported in the AL-KHWUIER_33_3G site and observed by the NOC surveillance team. The network performance team immediately informed the back office engineer to resolve the issue, as shown in Figure 7.1.

- The back office engineer checks the traffic behavior on the site to observe the number of users served by the site over a specific period, since high traffic could cause this issue, as is shown in Figure 7.2.

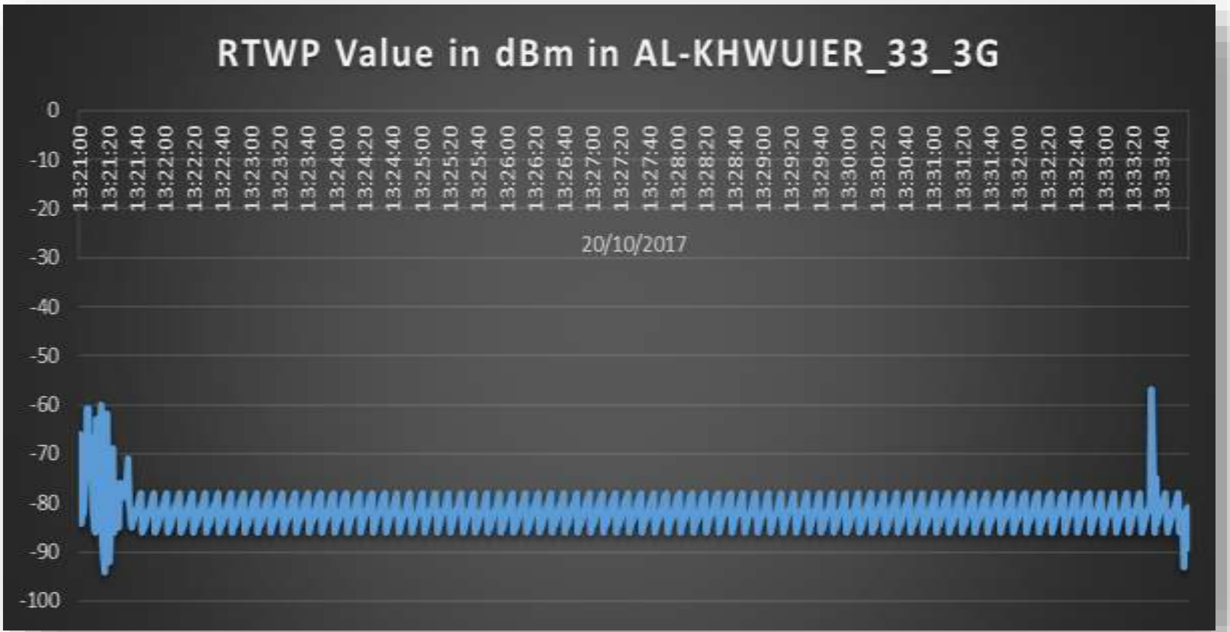


Figure 7.1 RTWP value in AL-KHWUIER_33_3G

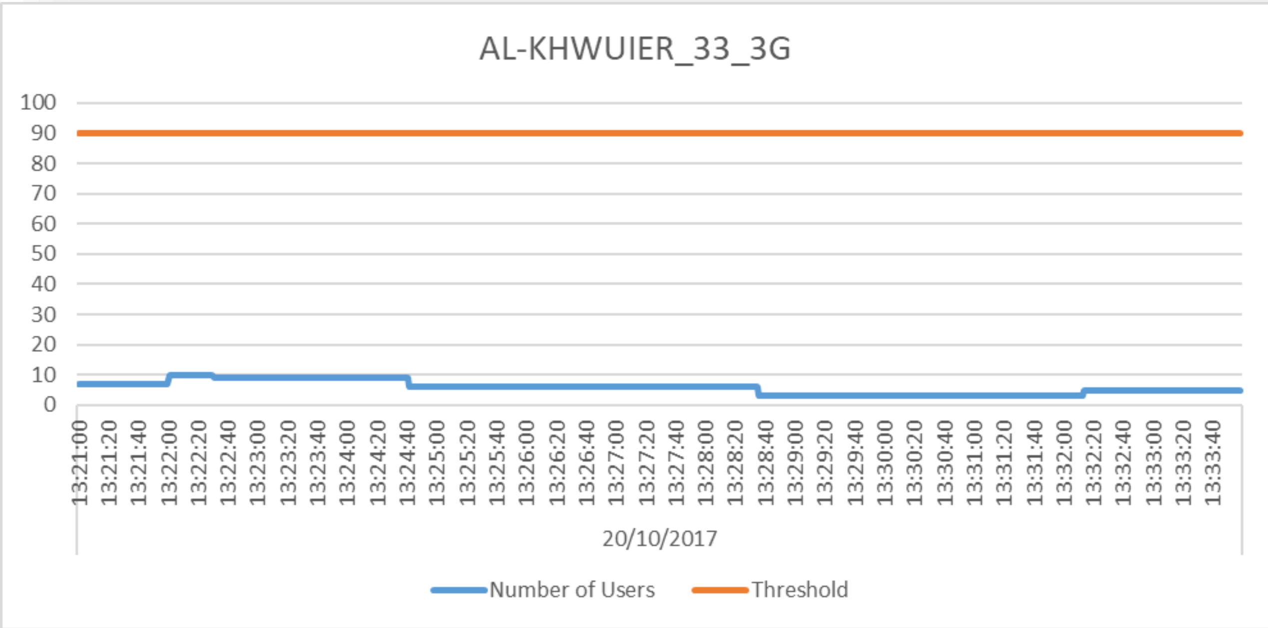


Figure 7.2 Number of users on the site during the alarm

- The back office engineer excludes whether it is high traffic that could cause the RTWP issue in the site. The back office engineer checks to see if there is any issue on the NodeB boards by checking the boards' status. Figure 7.3 shows the status of the boards in the NodeB.

```
DSP BRD:;
AL-KHWUIER_33_3G
+++ 77P5416GSNO 2017-10-20 13:05:21
O&M #38644
%%/*21227*/DSP BRD:;%%
RETCODE = 0 Succeed.
Board Status
------------
Cabinet No. Subrack No. Slot No. Board Type Standby Status Availability Status Operate State Admin State

Master          0        3        WBBP      NULL          Normal        Enabled      Unblocked
Master          0        7        WMPT      Active        Normal        Enabled      NULL
Master          0        16       UBF       NULL          Normal        Enabled      NULL
Master          0        19       UPEA      NULL          Normal        Enabled      NULL
Master          4        0        WRFU      NULL          Normal        Enabled      Unblocked
Master          4        2        WRFU      NULL          Normal        Enabled      Unblocked
Master          4        4        WRFU      NULL          Normal        Enabled      Unblocked
(Number of results = 7)
```

Figure 7.3 Board status in Huawei NodeB

- The back office engineer checks the configuration of the NodeB with regard to cell configuration and TMA, and in this case has found that the attenuation value was not properly configured (see Figure 7.4).

```
LST RXBRANCH:;
AL-KHWUIER_33_3G
+++    AL-KHWUIER_33_3G        2017-10-20 13:30:52
O&M    #21067
%%/*128897098*/LST RXBRANCH:;%%
RETCODE = 0  Operation succeeded.

List RxBranch Configure Information
-----------------------------------
Cabinet No.  Subrack No.  Slot No.  RX Channel No.  Logical Switch of RX Channel  Attenuation(

0            4            0         0               ON                            0
0            4            0         1               ON                            0
0            4            2         0               ON                            0
0            4            2         1               ON                            0
0            4            4         0               ON                            0
0            4            4         1               ON                            0

(Number of results = 6)
```

Figure 7.4 RX branch result before the troubleshooting

The back office engineer changes the configuration of the RX attention from 0 to 16 dbm. Figure 7.5 illustrates the operation for checking RX attenuation and changes it from 0 to 16 in Huawei NodeB 3900.

```
LST RXBRANCH:;
AL-KHWUIER_33_3G
+++    AL-KHWUIER_33_3G        2017-10-20 13:30:52
O&M    #21067
%%/*128897098*/LST RXBRANCH:;%%
RETCODE = 0  Operation succeeded.

List RxBranch Configure Information
-----------------------------------
Cabinet No.  Subrack No.  Slot No.  RX Channel No.  Logical Switch of RX Channel  Attenuation(0.5dB)

0            4            0         0               ON                            0
0            4            0         1               ON                            0
0            4            2         0               ON                            0
0            4            2         1               ON                            0
0            4            4         0               ON                            0
0            4            4         1               ON                            0
```

```
MOD RXBRANCH:CN=0,SRN=4,SN=0,RXNO=0,ATTEN=16;
AL-KHWUIER_33_3G
+++     AL-KHWUIER_33_3G          2017-10-20 13:31:23
O&M     #21069
%%/*128897098*/MOD RXBRANCH:CN=0,SRN=4,SN=0,RXNO=0,ATTEN=16;%%
RETCODE = 0  Operation succeeded.
MOD RXBRANCH:CN=0,SRN=4,SN=2,RXNO=0,ATTEN=16;
AL-KHWUIER_33_3G
+++     AL-KHWUIER_33_3G          2017-10-20 13:31:24
O&M     #21070
%%/*12887099*/MOD RXBRANCH:CN=0,SRN=4,SN=0,RXNO=1,ATTEN=16;%%

MOD RXBRANCH:CN=0,SRN=4,SN=2,RXNO=0,ATTEN=16;
AL-KHWUIER_33_3G
+++     AL-KHWUIER_33_3G          2017-10-20 13:31:25
O&M     #21071
%%/*128897100*/MOD RXBRANCH:CN=0,SRN=4,SN=2,RXNO=0,ATTEN=16;%%
RETCODE = 0  Operation succeeded.
MOD RXBRANCH:CN=0,SRN=4,SN=2,RXNO=1,ATTEN=16;
AL-KHWUIER_33_3G
+++     AL-KHWUIER_33_3G          2017-10-20 13:31:26
O&M     #21072
%%/*128897101*/MOD RXBRANCH:CN=0,SRN=4,SN=2,RXNO=1,ATTEN=16;%%
RETCODE = 0  Operation succeeded.

MOD RXBRANCH:CN=0,SRN=4,SN=4,RXNO=0,ATTEN=16;
AL-KHWUIER_33_3G
+++     AL-KHWUIER_33_3G          2017-10-20 13:31:27
O&M     #21073
%%/*128897102*/MOD RXBRANCH:CN=0,SRN=4,SN=4,RXNO=0,ATTEN=16;%%
```

```
LST RXBRANCH:;
AL-KHWUIER_33_3G
+++    AL-KHWUIER_33_3G        2017-10-20 13:33:54
O&M    #21074
%%/*128897114*/LST RXBRANCH:;%%
RETCODE = 0  Operation succeeded.

List RxBranch Configure Information
-----------------------------------
```

Cabinet No.	Subrack No.	Slot No.	RX Channel No.	Logical Switch of RX Channel	Attenuation(0.5dB)
0	4	0	0	ON	16
0	4	0	1	ON	16
0	4	2	0	ON	16
0	4	2	1	ON	16
0	4	4	0	ON	16
0	4	4	1	ON	16

```
(Number of results = 6)
```

Figure 7.5 RX branch setting operation

Once the above operation has been executed successfully, the back office engineer starts to monitor the KPI of the affected site, noticing that the high RTWP value is normal after setting the new value for the RX attenuation. Figure 6.6 shows the improvement noticed in the high RTWP KPI.

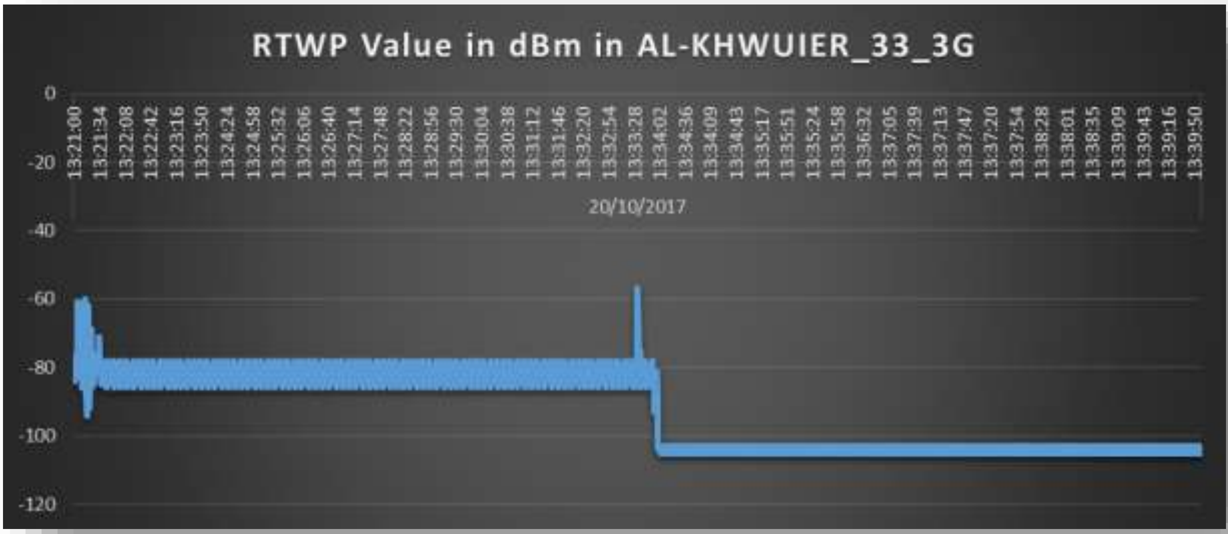


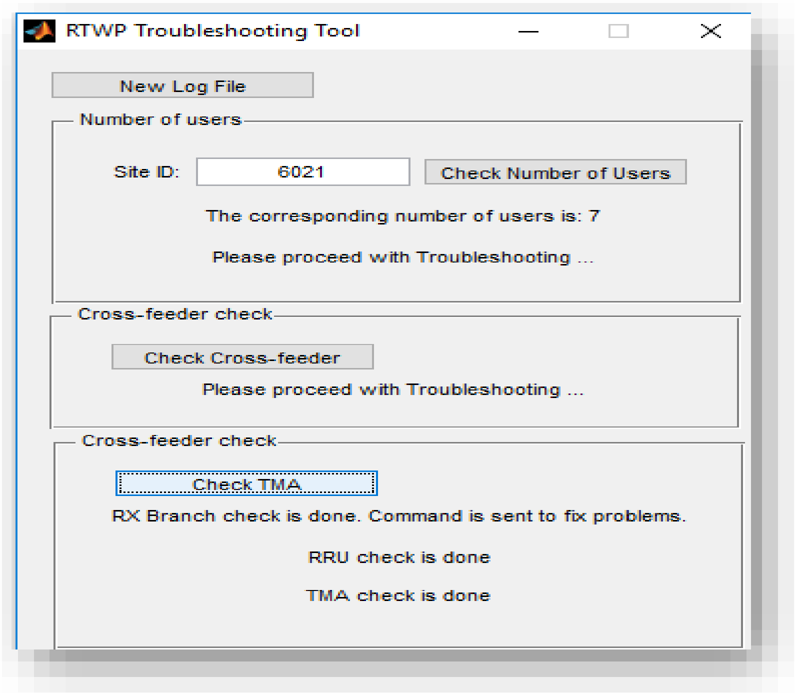
Figure 7.6 RTWP back to normal range

7.3 First Scenario Using the GUI Tool

In this section, the back office RAN engineer considered the same scenario as in Section 7.2, but using the GUI tool, which is designed to assist him in resolving the RTWP issue in a single site. The engineer who carried out this test thus interacted with the tool, which receives and generates the same commands as the Huawei NodeB 3900 standard. The back office RAN engineer logged in to the tool and performed the following actions to resolve the issue on the affected site:

- Entered the site ID in the tool to check whether the site was overloaded with a huge number of users; he found that the average number of users on the site per second was not exceeding 7 users per second.
- Checked the hardware of the site and crossed feeder connections, did not find any abnormality in the hardware or software-related part, and thus continued with the troubleshooting.
- After checking the RX, TMA and RRU configuration, he checked the site and the tool, and found that the RX attenuation was not properly adjusted; he generated the correct command to be executed in the NodeB.

Figure 7.7 shows all the steps, which explain the whole process of resolving the issue on the site, starting from using the tool till the command generated along with the operation log, to show how the tool can improve the efficiency of troubleshooting the RTWP alarm.



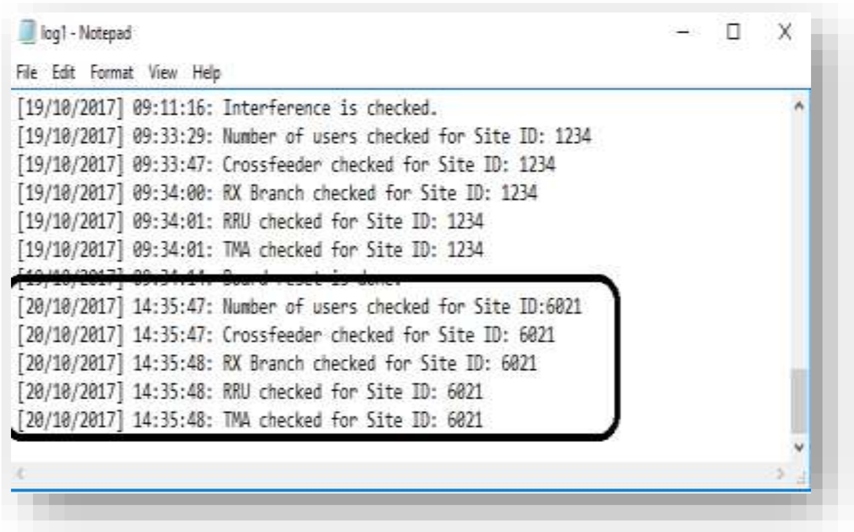


Figure 7.7 Whole process using tool in single-site mode

This section moreover contains a comparison between the manual approach of troubleshooting and using the Interface GUI tool for a single site, starting from checking the number of users, checking the hardware configuration, and finding the missing configuration in the RX branch, till executing the commands in the Huawei NodeB 3900. Figure 7.8 illustrates this comparison in terms of time taken to execute the commands in Huawei NodeB 3900. As mentioned in Section 7.1, the skill of the engineer who is handling the issue plays an important role in troubleshooting. Inexperienced engineers need some time to read the solution manual of the alarm and to decide what to do. This test was performed by an experienced RAN back office engineer.

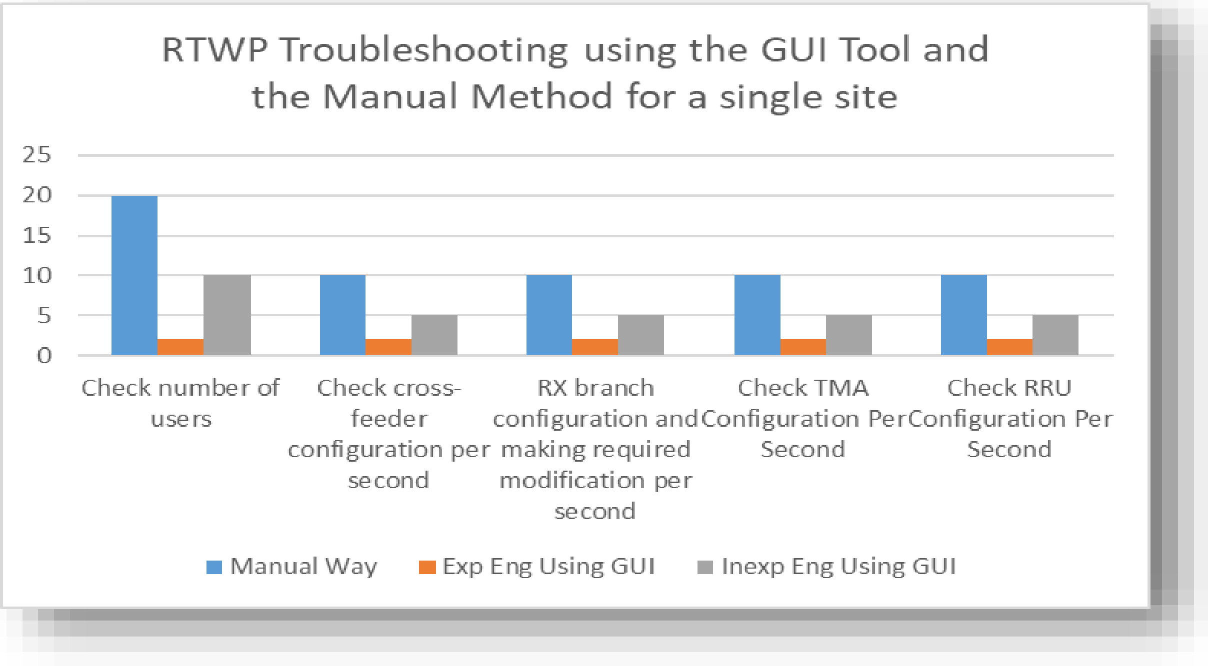


Figure 7.8 Time difference between GUI and Manual approach

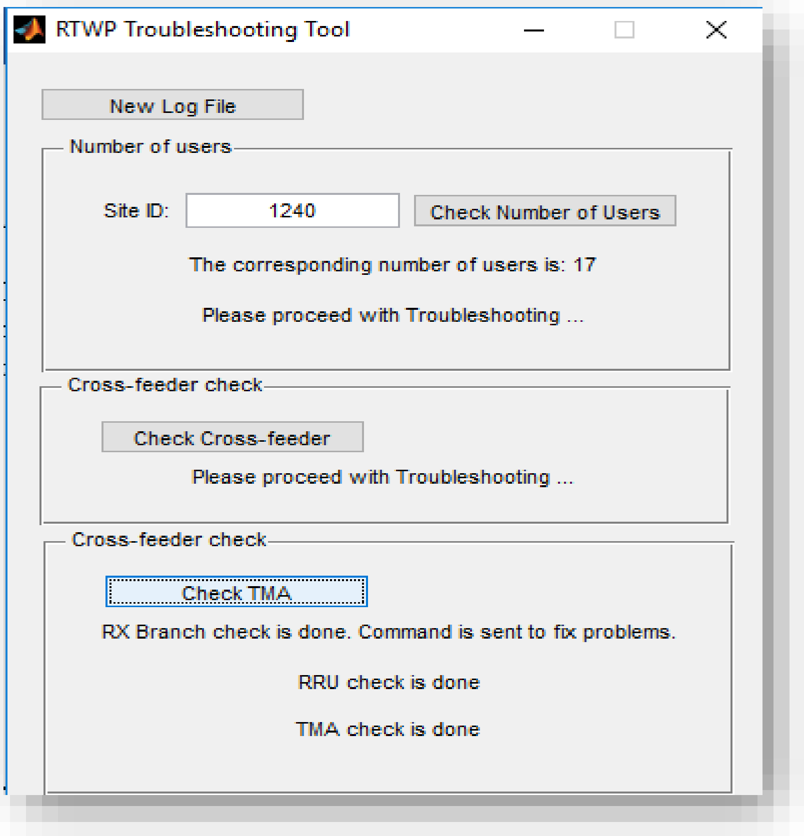
The above bar chart compares both approaches. It is clear that the tool is more effective and efficient in terms of time. Each step is quicker and can be performed by the tool within only 2 seconds, whereas the manual approach takes much longer, especially if the engineer is not experienced. In addition, there is no margin of error when using the tool. Each step is more than 80% quicker when using the GUI than when using the manual approach.

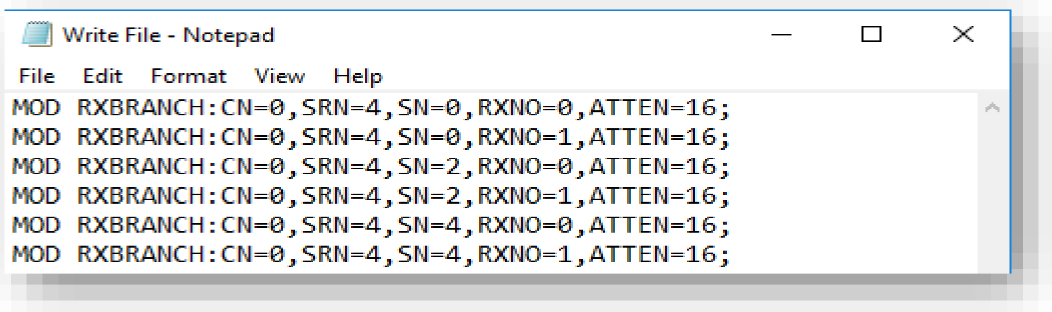
In order to get a good comparison, an inexperienced engineer used the GUI tool to solve the RTWP issue on a single site, viz. DFGH with site 1240. The engineer logged in to the tool and performed the following actions to resolve the issue in the affected site:

- Entered the site ID to check the number of users served by the site.

- Checked the hardware of the site and of any crossed feeder connections, did not find any abnormality in the hardware or software-related part, and continued with troubleshooting.

- The engineer proceeds with troubleshooting by checking the RX branch configuration; the GUI tool showed that some misconfiguration was found in the NodeB. The generated commands had to be executed in the NodeB to fix the RTWP issue. Figure 7.9 shows the messages and interactions with the tool, as well as the commands generated by the tool to fix the issue and the operation log.





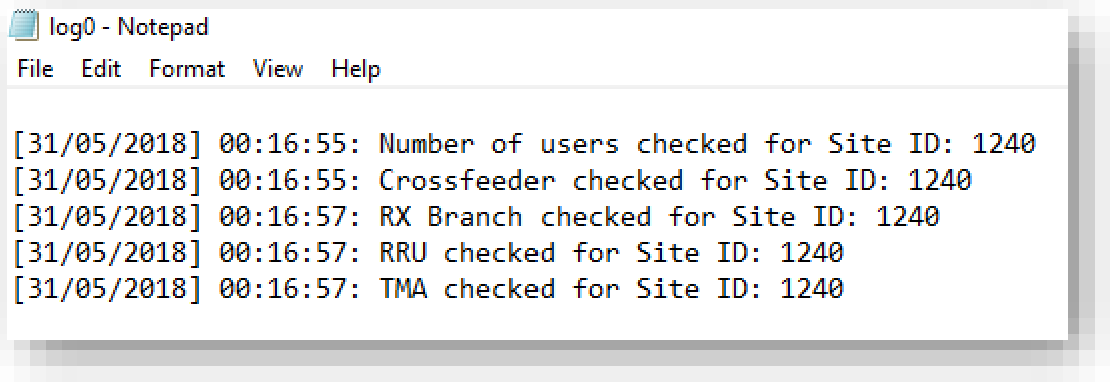


Figure 7.9 Operation by inexperienced engineer

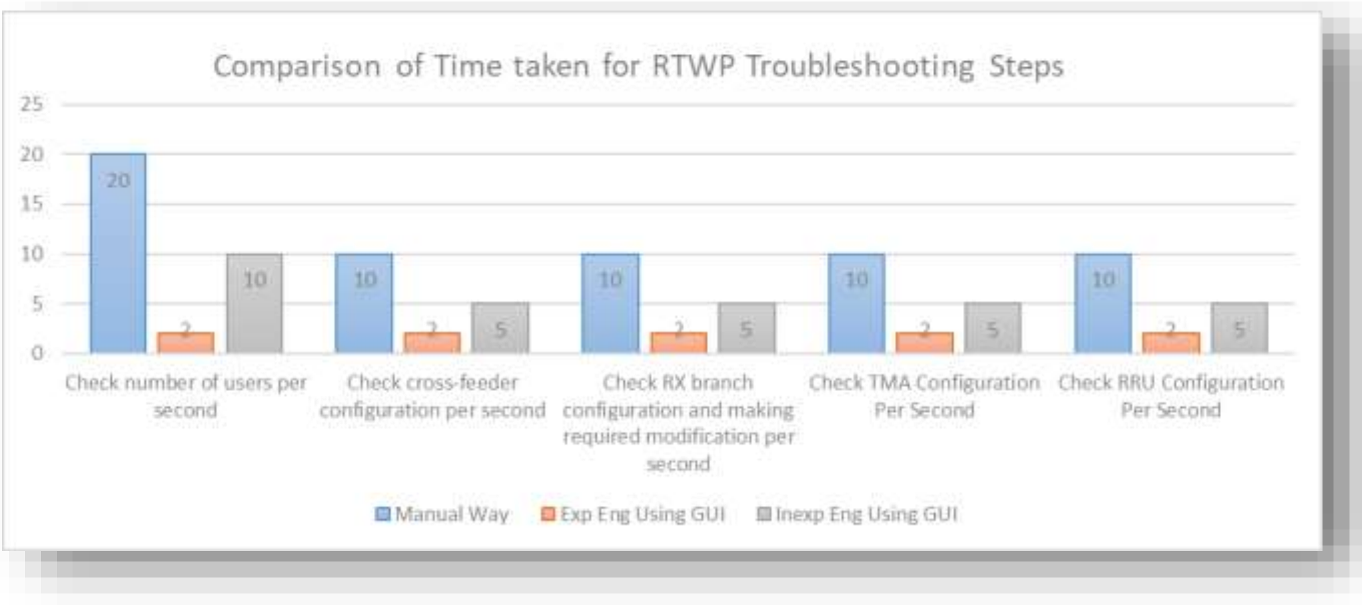


Figure 7.10 Inexperienced engineer using the GUI with other steps

The above bar chart compares three steps used in troubleshooting, viz. the manual approach, the use of the GUI tool by an experienced engineer and the use of the GUI tool by an inexperienced engineer. When it was used by inexperienced engineer, the time taken for each step was 50% less than in the manual approach, which means that the efficiency of troubleshooting was also improved in this case.

7.4 Second Scenario

In the second scenario, it was tested whether the GUI tool would help the back office engineer to check the interference status and notify the RF team about the issue. The NOC noticed two sites suffering from a high number of RTWP errors, and the surveillance team suspected an interference issue on both sites. The problem was escalated to the RAN back office engineer. By checking the value of the RTWP at both sites, it was found to be abnormal, as illustrated in Figure 7.11. The average number of users per site was normal and not exceeding 12 users per second, as illustrated in Figure 7.12.

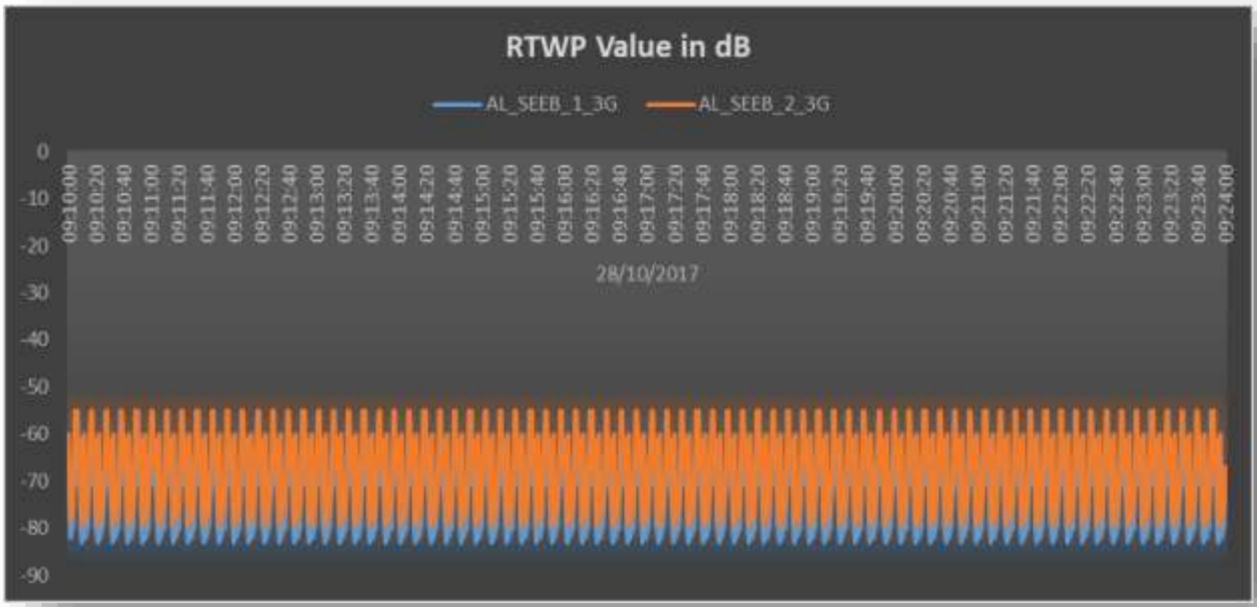


Figure 7.11 RTWP values on both sites

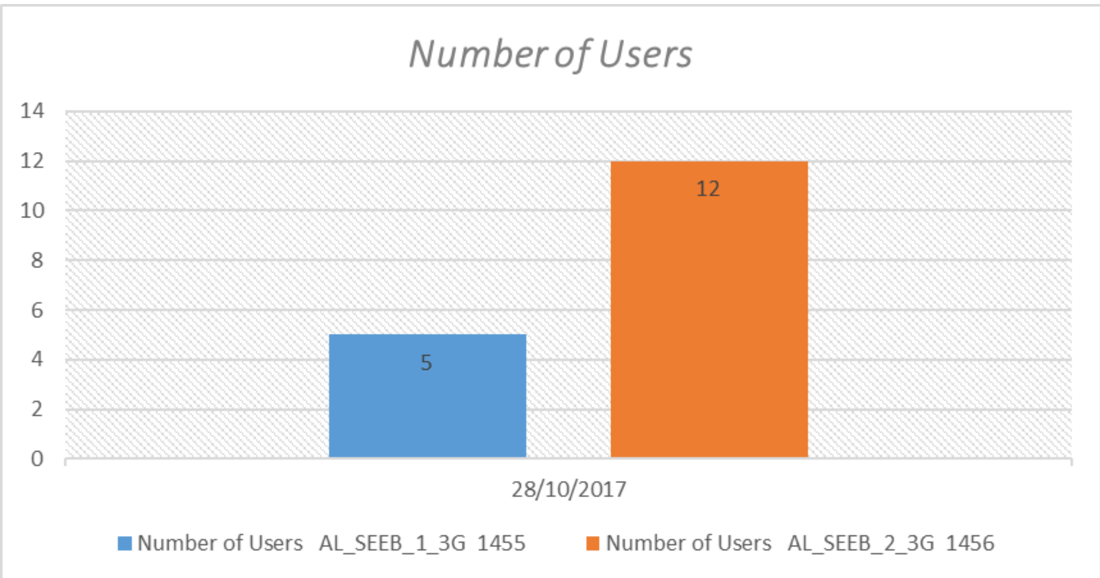
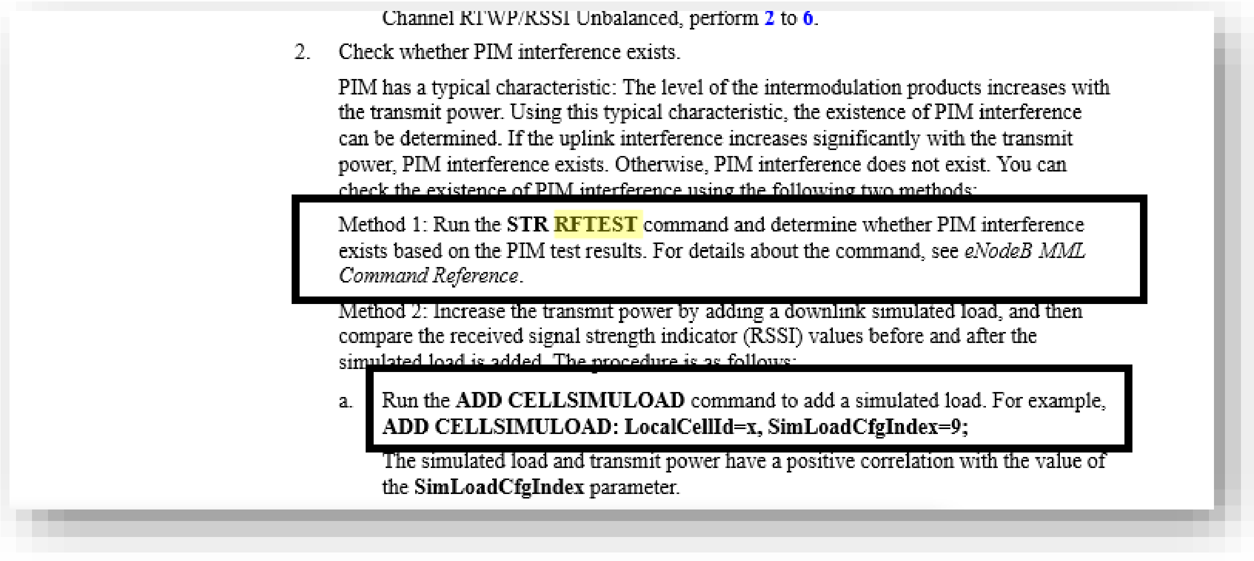


Figure 7.12 Number of users on both sites

The RAN back office engineer in this scenario needed to perform two actions at both sites, according to the Huawei NodeB 3900 operation manual: he needed to test the interface and increase the TX power (Figure 7.13). This operation is not allowed to be executed in daytime or busy hours, however; it can only be performed during off-peak hours, such as after midnight, since it has a huge impact on the KPI.


Channel RTWP/RSSI Unbalanced, perform 2 to 6.

2. Check whether PIM interference exists.

PIM has a typical characteristic: The level of the intermodulation products increases with the transmit power. Using this typical characteristic, the existence of PIM interference can be determined. If the uplink interference increases significantly with the transmit power, PIM interference exists. Otherwise, PIM interference does not exist. You can check the existence of PIM interference using the following two methods:

Method 1: Run the **STR RFTEST** command and determine whether PIM interference exists based on the PIM test results. For details about the command, see *eNodeB MML Command Reference*.

Method 2: Increase the transmit power by adding a downlink simulated load, and then compare the received signal strength indicator (RSSI) values before and after the simulated load is added. The procedure is as follows:

a. Run the **ADD CELLSIMULOAD** command to add a simulated load. For example, **ADD CELLSIMULOAD: LocalCellId=x, SimLoadCfgIndex=9;**

The simulated load and transmit power have a positive correlation with the value of the **SimLoadCfgIndex** parameter.

Figure 7.13 Operation document for Huawei NodeB 3900

The GUI tool is envisaged to help the RAN back office engineer to perform this operation, starting from checking the number of users through to sending the issue to the RF team to investigate on site, if the issue cannot be resolved through the remote connection.

- The engineer imports the configuration files that contain the relevant cell configurations. The tool then presents the number of users to the engineer (Figure 7.14).

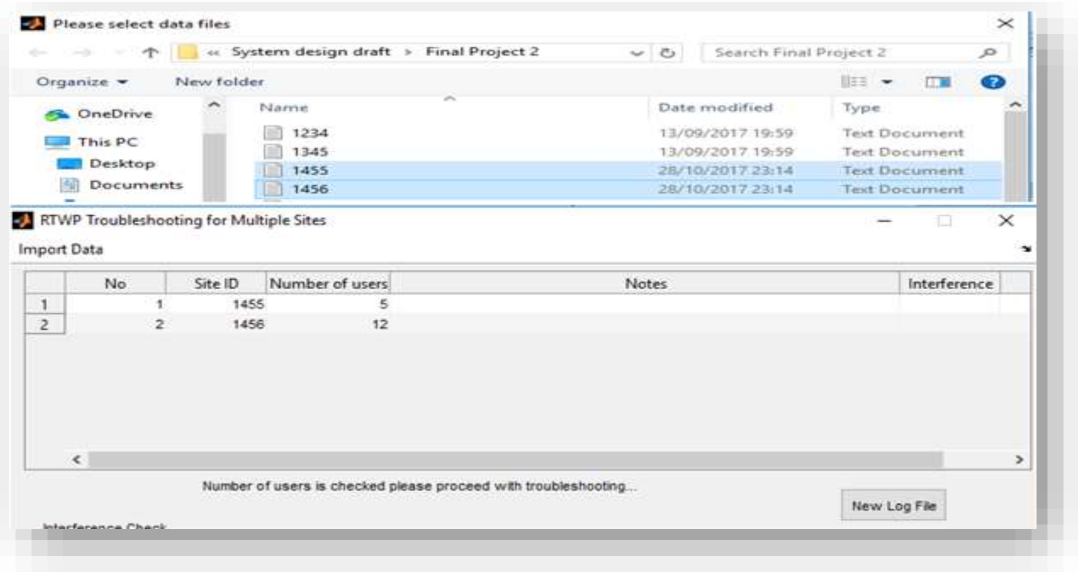


Figure 7.14 Importing NodeB configuration files and determining the number of users

- In the next step, the user needs to check the interference status on both sites by executing "STR RFTEST", which means starting the Radio Frequency test in Huawei NodeB 3900. The outcome of this execution from Huawei NodeB 3900 is either normal or abnormal. If the status is 'normal', there is no interference. A status of 'abnormal' means there is interference in NodeB; in this scenario, the status is abnormal.

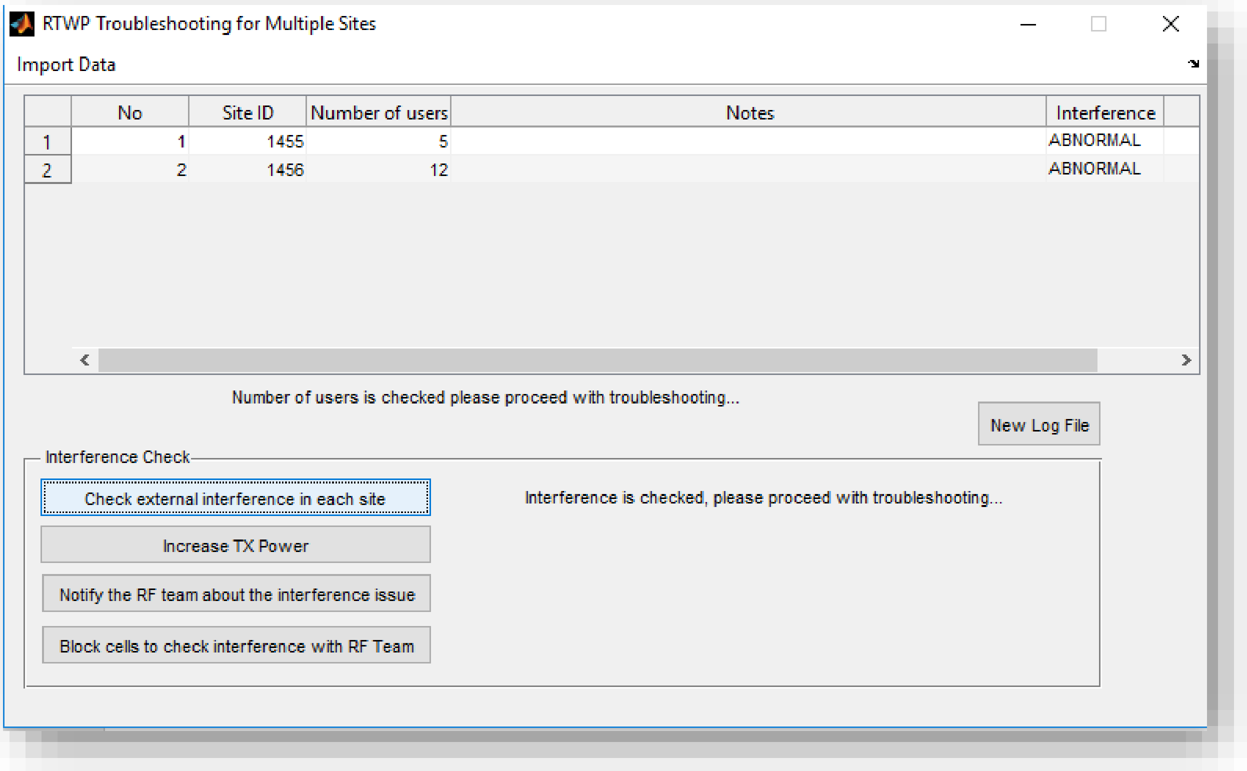


Figure 7.15 Output of the interference check

- The tool presents the output of an interference check on both sites and suggests proceeding with troubleshooting.

- In the next step, the engineer needs to execute "ADD CELLSIMULOAD" to simulate load and transmitting power. This command is generated automatically once the user clicks on this button. The tool will generate commands for each cell at both sites. Figure 7.16 and Figure 7.17 show the commands that are generated by the tool, together with the operation log produced when the commands are generated by the tool.

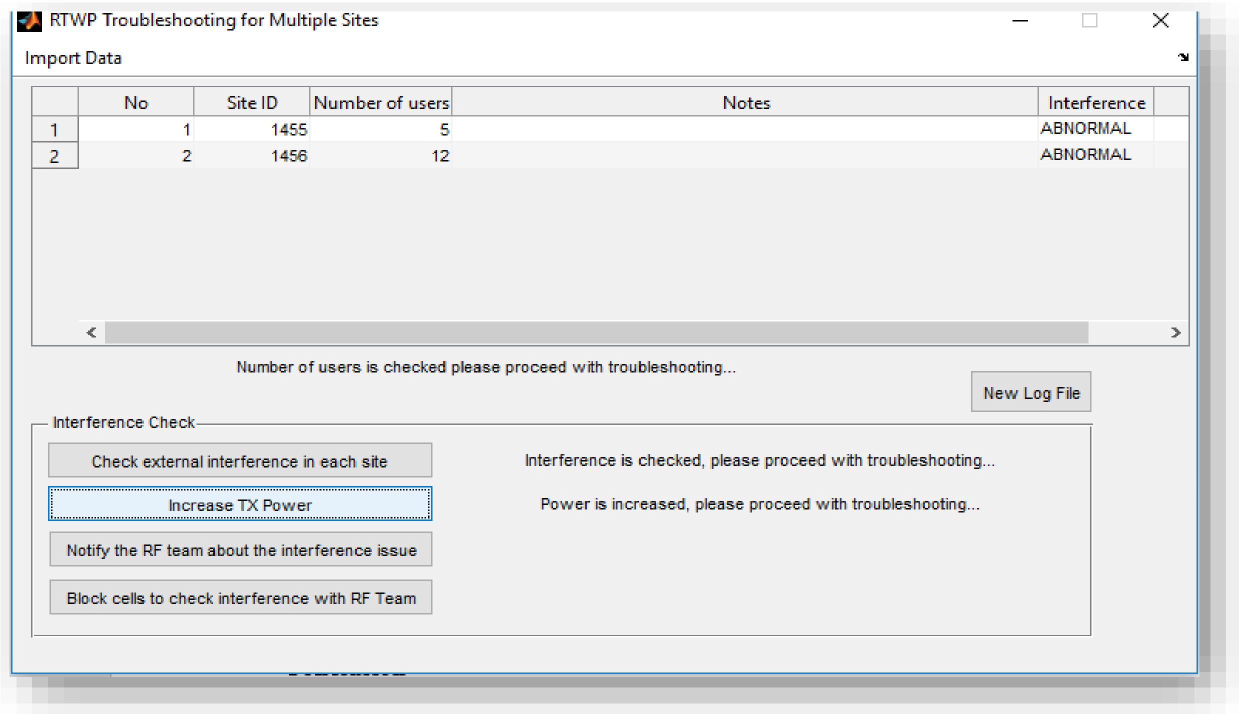


Figure 7.16 User click on Increase TX Power

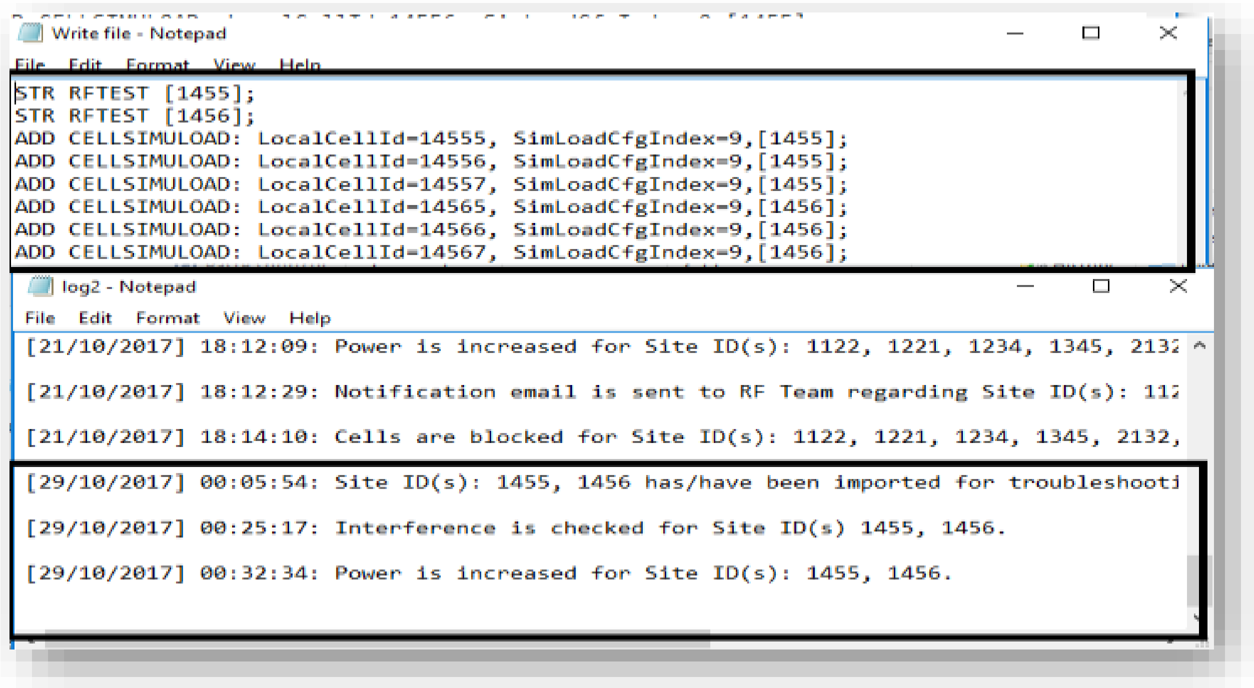


Figure 7.17 Command generated and operation log

- In the live network (i.e. in a real situation) the RAN back office engineer should execute these commands in NodeB at off-peak hours and start monitoring RTWP KPI after this operation. If no improvement is observed, then the issue will need to be handled by the

RF team to identify the sources of interface. The tool will help with this by sending an automatically generated email to the RF team when the user clicks on the 'Notify RF Team' button to inform the RF team about the interference. The RF team then needs to visit the sites with a spectrum analyser to identify the sources of interference. Once the RF team arrives on site, the back office engineer needs to block the cells, thus allowing the RF team to hunt for the source of the interference. Figure 7.18 illustrates these operations with email and block commands for blocking the services in the cells, as shown in Figure 7.19.

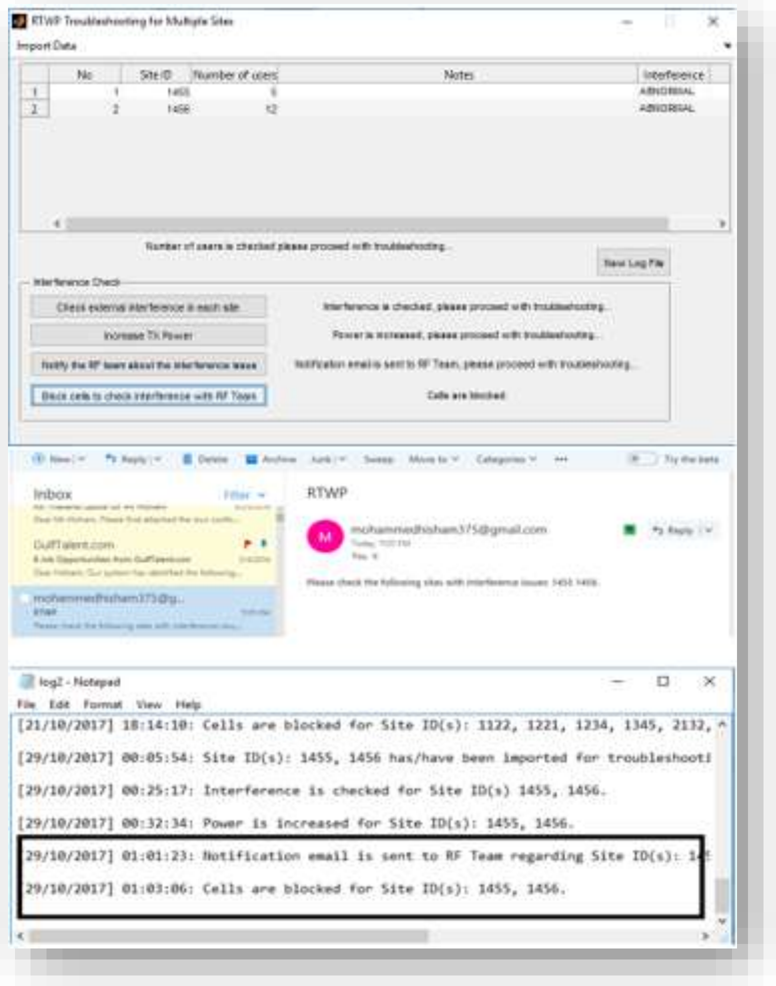


Figure 7.18 Email notification with operation logs

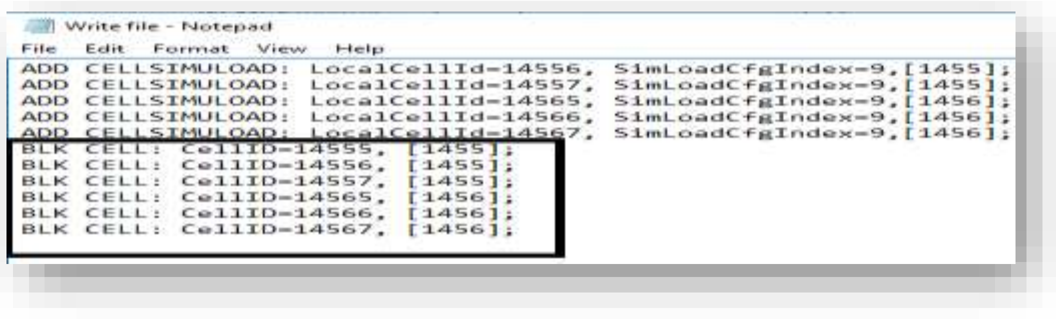


Figure 7.19 Block cells command

- Once the service is blocked in the cell by the above command, the RF team needs to conduct several tests in the area to identify the possible source of interference. The team needs to be equipped with a spectrum analyser, a frequency scanner, a GPS and a camera. Once their task is completed, they then need to inform the radio planning team. Meanwhile the back office RAN engineer will unlock the services in the affected cells, irrespective of whether the interference is fully resolved, to maintain the provision of whatever service is possible, dependent on the remaining interference. The radio planning team will decide on the appropriate solution, for instance, to relocate the node in the affected area or to negotiate with the owner of the transmitter that is causing the interference about possibly adjusting the frequency of transmitting.

7.5 Third Scenario

The NOC surveillance team observed a high RTWP on two sites in the network at a busy hour and decided to send it to the back office RAN engineer for further troubleshooting. The back office engineer decided to use the GUI tool for diagnosis and troubleshooting directly. He thus imported the files into the tool and noticed that the number of users was below the threshold value and that interference was abnormal at both sites; the tool recommended to carry on troubleshooting. Figure 7.20 shows the number of users and the interference status. Since it was midnight, the engineer decided to increase the TX power in both NodeB and observed RTWP KPI after the execution.

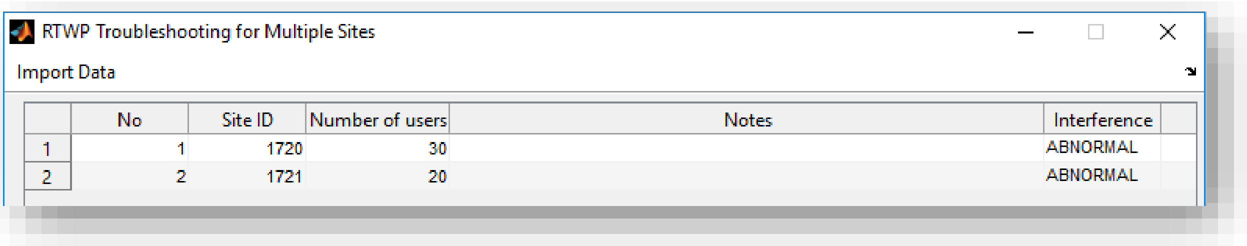


Figure 7.20 Number of users

- Figure 7.21 shows what happens when the back office engineer clicked the “increase TX power” button and the tool generates the required commands accordingly. Figure 7.22 shows the commands generated by the tool.

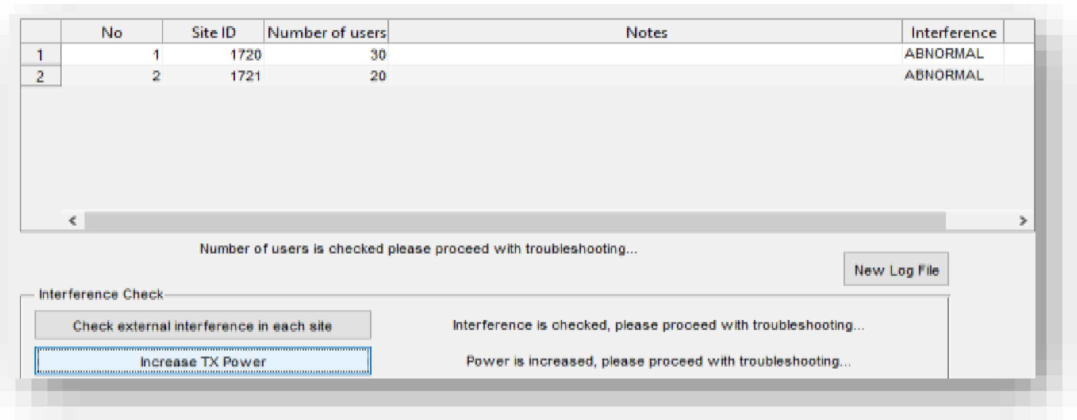


Figure 7.21 User clicks the “increase TX power” button

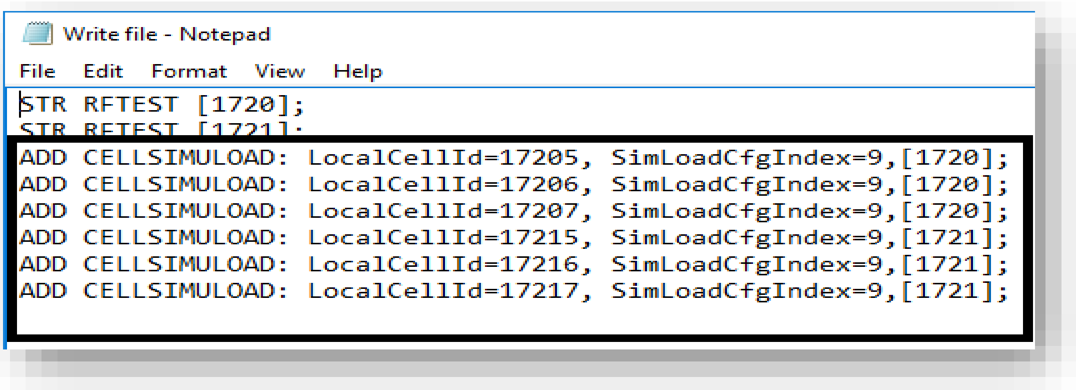


Figure 7.22 Commands generated to increase TX power

- The back office RAN engineer executed the command in the Huawei NodeB 3900 and started to observe the RTWP KPI after the execution. The RTWP reverted to the normal value, as illustrated in Figure 7.23 and Figure 7.24. The interference was eliminated from the sites after RTWP normalized.

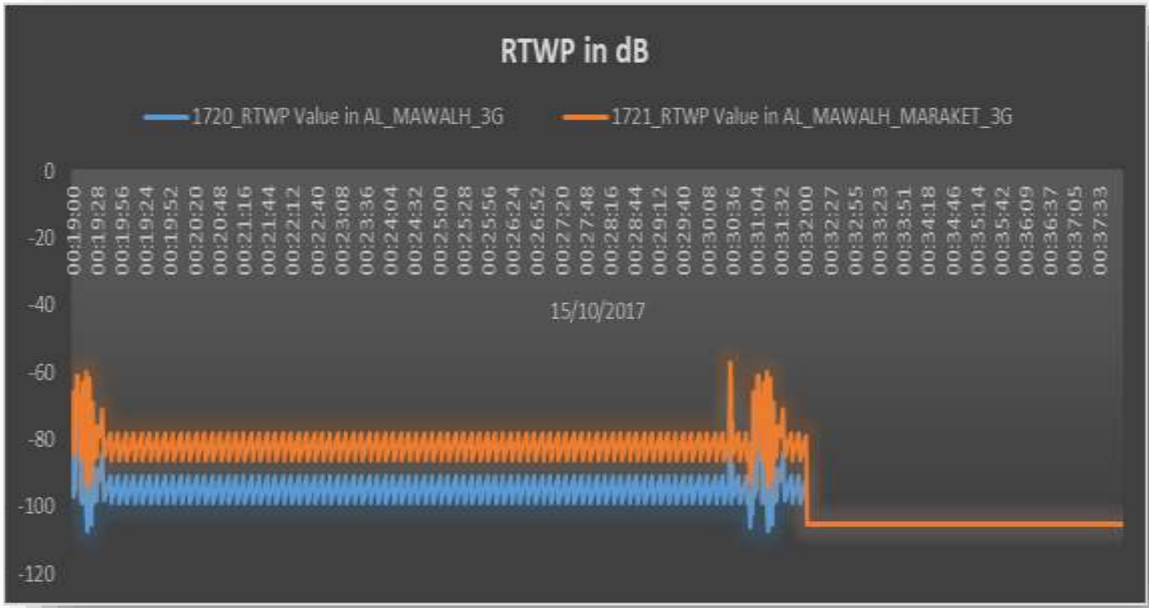


Figure 7.23 RWTP Normalized after TX power increased

	No	Site ID	Number of users	Notes	Interference
1	1	1720	8		NORMAL
2	2	1721	5		NORMAL

Figure 7.24 Interference normalized after the operation

7.6 Fourth Scenario

A high RTWP value observed in one site was escalated to the back office RAN engineer for further analysis and troubleshooting. The engineer used the GUI tool for diagnosis and analysis. He started by entering the site ID to see if there was high traffic in the site. The number of users was normal and within the capacity of Huawei NodeB 3900. The engineer thus carried on with troubleshooting by checking the NodeB configuration, as illustrated in Figure 7.25.

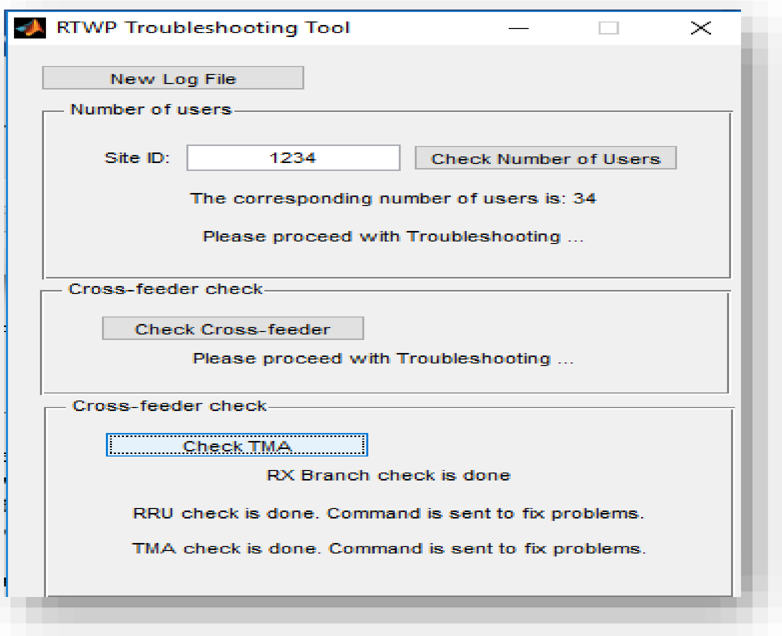
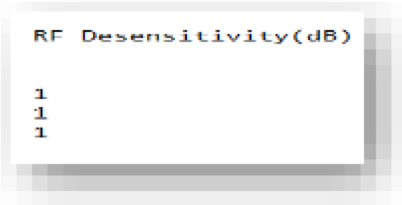
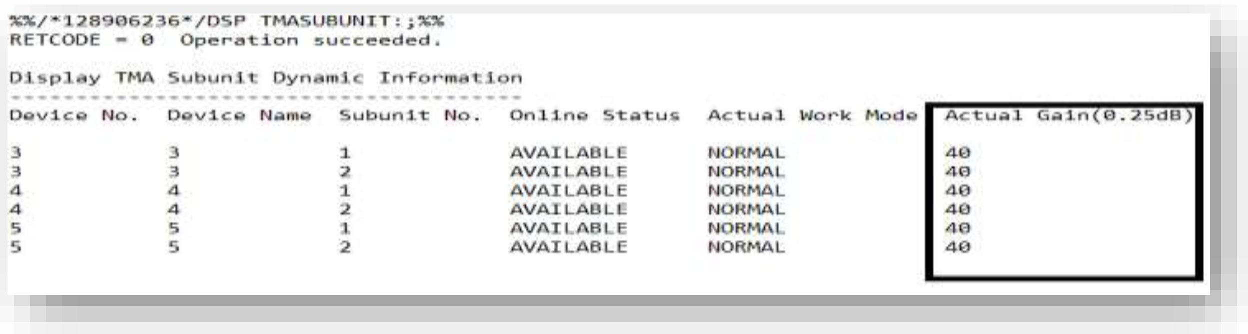


Figure 7.25 Configuration checking

- The tool showed the back office RAN engineer that there was a wrong configuration in the RRU and TMA. Accordingly, he generated the correct settings to resolve this problem, which were then executed. Figure 7.26 shows the commands generated, and the incorrect configuration found in the Huawei NodeB 3900 Read File. The 'RF de-sensitivity' setting was 1 instead of 0 and the 'gain' in TMA setting was 40 instead of 48. The tool generated a command based on this result.



RF Desensitivity(dB)
1
1
1



%%/*128906236*/DSP TMASUBUNIT:;%%
RETCODE = 0 Operation succeeded.

Display TMA Subunit Dynamic Information

Device No.	Device Name	Subunit No.	Online Status	Actual Work Mode	Actual Gain(0.25dB)
3	3	1	AVAILABLE	NORMAL	40
3	3	2	AVAILABLE	NORMAL	40
4	4	1	AVAILABLE	NORMAL	40
4	4	2	AVAILABLE	NORMAL	40
5	5	1	AVAILABLE	NORMAL	40
5	5	2	AVAILABLE	NORMAL	40

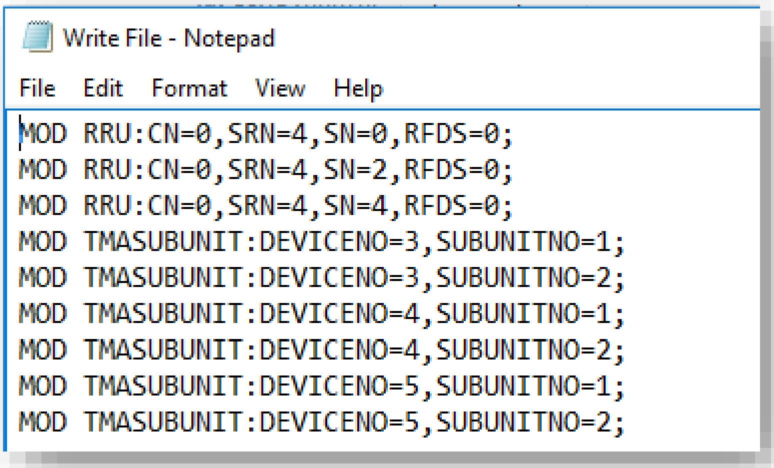


Figure 7.26 The incorrect NodeB configurations (top) and the commands that were generated to correct these settings (bottom).

- The back office RAN engineer then checked the RTWP value in the site to see whether these settings were the root cause of the issue or not. The RTWP KPI reverted to the normal range after the operation. Figure 7.27 shows that the value of the RTWP KPI was normalized after adjusting the wrong configurations in RRU and TMA.

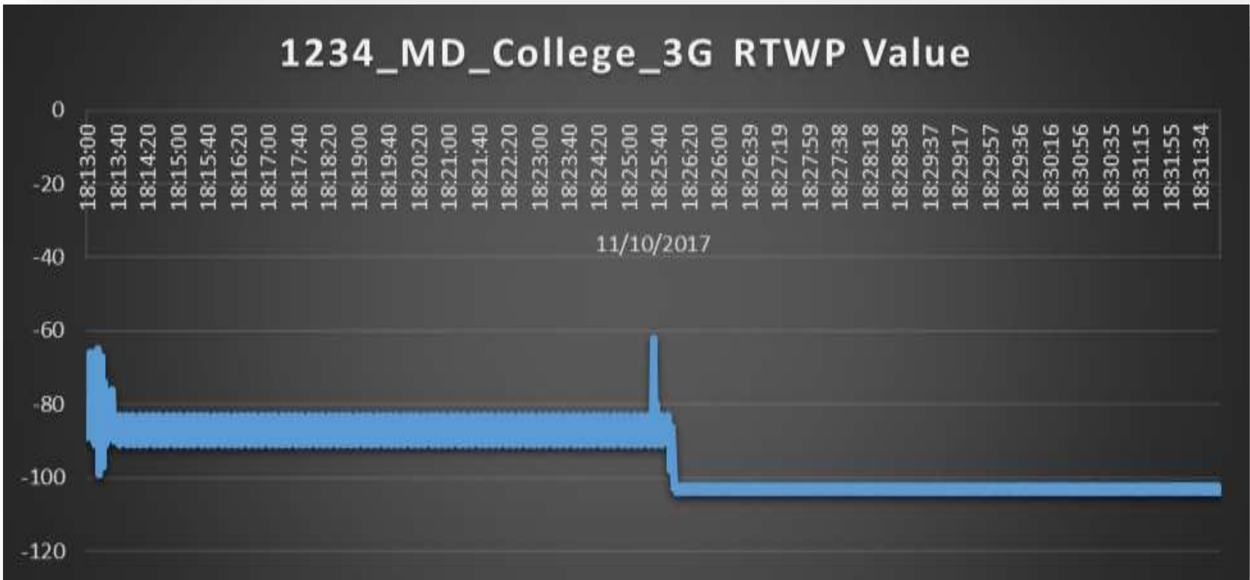


Figure 7.27 RWTP back to its normal value after changing RRU and TMA setting

7.7 Fifth Scenario

In the last scenario, the tool was used to diagnose several issues observed in a single site located in a hotspot area (i.e. an area of high use). An RTWP alarm was observed by the NOC during the busy hours of this area, and this problem was immediately managed by a back office RAN engineer. The engineer decided to use the tool during busy hours to check the root cause of the issue, as shown in Figure 7.28.

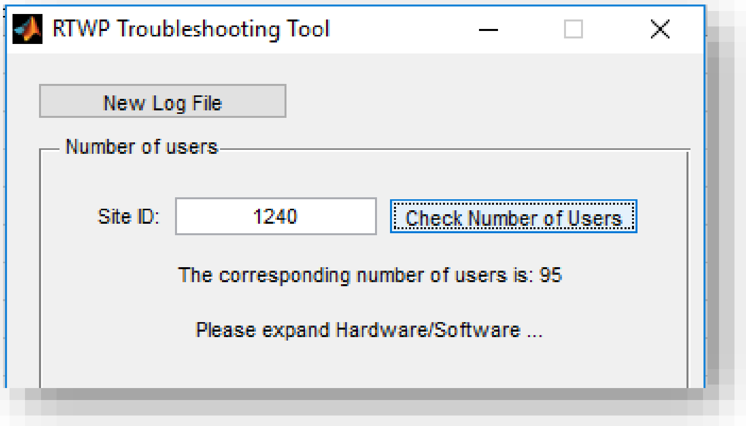


Figure 7.28 Message to expand the NodeB

The tool suggested expanding the hardware or software of the NodeB, identifying a high number of users observed in the site. The engineer decided to export the KPIs that were related to the number of users and the RTWP on the affected site. Figure 7.29 shows the output of these two KPIs

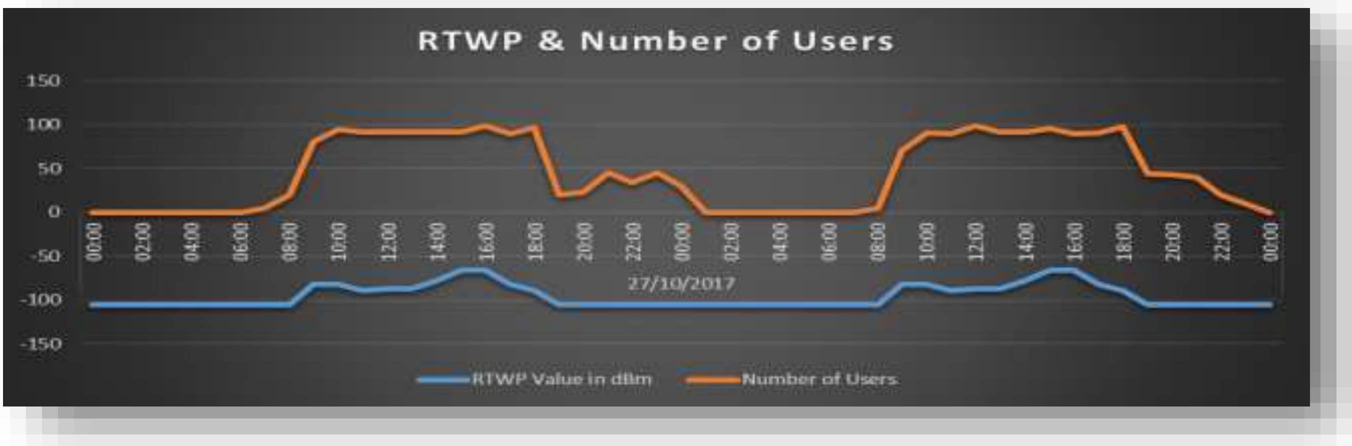


Figure 7.29 Number of users and RTWP

From the above figure, the engineer inferred that, whenever there was an increase in traffic, there would be a corresponding increase in RTWP, especially during the busy hours of the day. During the off-peak hours, however, when there were few users served by the NodeB, RTWP recorded a normal value; based on this analysis, the back office RAN engineer decided to escalate the issue to the Planning Department, requesting a capacity check for the NodeB. The Planning Department can expand the NodeB by means of hardware or software, depending on the availability of the hardware or software license.

7.8 Chapter Summary

This chapter contains a comparison between the manual approach of solving an RTWP issue in a single site setting and in multiple sites, and the various methods offered by using the GUI tool. The tool was used by both an experienced and an inexperienced back office RAN engineer to resolve the issue of RTWP errors in a 3G network. Time-consuming issues in both scenarios were presented to show the efficiency of the tool and the fact that it significantly reduced the time needed to troubleshoot both single-site and multi-site issues. The tests also demonstrated how the GUI tool can support the back office RAN engineer in situations where interference is affecting two sites, and how it can support the operation team when an RTWP issue needs an RF team to be deployed. This starts from checking the number of users on the site, conducting a frequency scan test, increasing the simulation load and TX power, and even escalating the problem to other departments, when a software solution is not effective, by sending email notifications and generating commands to block the services in a particular cell. Five scenarios were discussed in the chapter covering the most important aspect of the GUI tool, excluding two scenarios that were not tested (board and NodeB restart) . The outcomes from these tests and scenarios met the expectation and objectives of the study.

CHAPTER 8: CONCLUSION

8.1 Introduction

This chapter starts by reflecting on the problem statement presented at the start of this dissertation. Thereafter, the objectives are reviewed and conclusions are drawn concerning the overarching objectives of this dissertation, as set out in Chapter 1. Lastly, this chapter also discusses possible future extension of this work.

8.2 Discussion of Problem Statements

The proposed solution enables the NOC to diagnose and assist in solving RTWP issues and maintaining the target SLA by improving the efficiency of responses by maintenance team. The dissertation has discussed the root causes of RTWP errors and the general procedures that are followed by mobile network operators when they deal with interference issues in a live network. From the results, it has been proven that the GUI tool can help the operation team to diagnose the root causes of network problems, to check the relevant configurations, and to adjust an incorrect configuration automatically. The tool furthermore sends notifications in case the situation needs to be to be escalated to another department or to involve the RF team. Furthermore, when interference is having an impact on several sites, the tool can help the operation team to save time in checking the confirmation data and in determining the likely faults, based on automatically checking the configuration settings. In addition, logs are generated to assist the operations team in report writing and for billing purposes. The GUI tool enables the automated checking of parameters to be sent to the NodeB at a particular site. This tool was tested successfully by an experienced engineer (as well as an inexperienced engineer for the sake of comparison), and a comparison was made between the manual approaches of handling a fault and the procedures recommended by the tool for resolving an RTWP issue in a single site; the outcome of this comparison meets the objectives mentioned in Section 1.3, particularly with regard to reducing the time needed and improving the efficiency of troubleshooting and diagnosing problems at a site. The scenario for two sites impacted by interference were tested, following all the procedural stages through to the situation where an RF team had to be dispatched to the site for further analysis and manual intervention to correct the problem. The research objects mentioned in Section 1.3 were thus achieved, and they are summarized in the following points:

- The manual process was automated successfully and implemented: it was shown in Section 6.2 and Section 6.3 how each functionality of the GUI tool was demonstrated, starting from checking the numbers of users, all the way to the point of sending an email notification if the RF team needed to be sent to the site. The GUI tool successfully checked the configuration, and generated commands to assist in resolving the various RTWP issues.

- Time minimization and increased efficiency were proven during the testing. In Section 7.1 and Section 7.2 the manual and automated approaches of clearing RTWP errors in a single site were tested, respectively. It was shown that the time taken for each troubleshooting step was reduced by 80% when the GUI tool was used (see Figure 7.8).

- The GUI tool can be used by non-experienced engineer to resolve RTWP issues: Section 3.5 contains a help guide that can assist the inexperienced engineer to understand the functionality of each button. In addition, the sequence of steps supported by the GUI tool is explained, which will assist the inexperienced engineer to carry out the tasks that would be needed to clear a RTWP alarm condition for either single or multiple sites.

- The GUI tool can be modified according to new requirements. In Section 3.9, it was described how the GUI tool could be adjusted to support an alternate NodeB vendor. An example of modifying the relevant MATLAB code, to support Ericsson NodeB, was presented in this section.

8.3 Future Work

The GUI tool can be modified, as mentioned in the methodology chapter (see Section 3.9), to accommodate future enhancements or to support other vendors of the NodeB system, such as Ericsson [38]. Changing the relevant commands from the Huawei NodeB 3900 to the Ericsson NodeB was examined in Section 3.9. However, in the future this could be expanded to accommodate all the functional aspects of the GUI tool.

The tool could be redesigned to operate in a more autonomous manner, which could be utilized as a fundamental component of developing a self-healing SON network.

This tool could furthermore allow for the operation of the SON to perform all the regular maintenance and troubleshooting operations without any human interaction. The tool could be designed to learn from prior fault-finding techniques, and to learn from these experiences how to resolve recurring problems more quickly. The troubleshooting concepts could be integrated into a SON that can spread troubleshooting techniques to different parts of the network, to better configure, tune, optimize and heal the cellular system issue more rapidly. Self-healing is defined as a SON procedure that detects a problem and mitigates it without human interaction. The two major areas where the self-healing concept could be applied are as follows: [30].

- Self-diagnosis: create a model to diagnose, learning from past experiences.
- Self-healing: automatically start the corrective actions to solve the problem.

Furthermore, a new feature can be added in the tool to check different issues, for example, Remote Electrical Tilt (RET) alarms by adding a new button in the tool and adding the RET configuration in the MATLAB code.

8.4 Chapter Summary

This chapter has given a brief overview of what this dissertation has achieved and highlighted the effectiveness of the tool that was developed to support solving RTWP issues. Future modifications that could be made to the proposed tool were also presented. In summary, the objectives that were set out at the start of the dissertation have been achieved, and the required GUI tool has been implemented and tested in a selection of representative scenarios.

REFERENCES

[1] Yanjiao Chen, Lingjie Duan, Qian Zhang, “Financial Analysis of 4G Network Deployment”, IEEE Conference on Computer Communications (INFOCOM), 2015.

[2] Oman Telecommunication Network Operation Centre (NOC) Database/RTWP Process, NOC, Shiraz, 27-08-2011.

[3] Zhang Junjie, Liu Qiong, “Huawei RTWP Troubleshooting Guide”, available: https://www.scribd.com/doc/174672142/RTWP-Troubleshooting, accessed on 25-March-2017

[4] Zhenxian Hu, Yi-Chao Chen, Lili Qiu, Guangtao Xue, Hongzi Zhu, Nicholas Zhang, Cheng He, Lujia Pan, “An In-depth Analysis of 3G Traffic and Performance’’, Shanghai Jiao Tong University, The University of Texas at Austin, Huawei, Aug 17, 2015

[5] Leopedrini, “TelcomHall”, available: www.telecomhall.com/what-is-rtwp.aspx, accessed 12-July-2017

[6] Viavi, “Interference case studies’’ available: www.viavisolutions.com/en-us/literature/high-rtwp-problem-solved-interferenceadvisor-case-studies-en.pdf, accessed on 11-June-2017.

[7] Agilent Technology, “Interference Testing with Handheld Spectrum Analyzers’’, Published in USA, October 11, 2011 5990-9074EN, available: www.ccontrols.ch/cms/upload/applikationen/LTE/5990-9074EN, accessed on 07-October-2017.

[8] Huawei, “DBS3900 commission procedure’’ available: www.academia.edu/6492331/Huawei_DBS3900_Commissioning_Procedure_Huawei_DBS3900_Commissioning_Pro, accessed on 28-10-2017.

[9] Huawei, “Huawei NodeB V200R013 Technical Description’’, available on https://cosconor.fr/GSM/Divers/Equipment/Huawei/NodeB%20technical%20description.pdf, accessed on 15-10-2017.

[10] Shivani Acharya1, Trilok Gaba, “A Review for Handover in 3G/UMTS Network’’, International journal of Emerging Trends in Science and Technology, 2014.

[11] Merriam Dictionary, available on https://www.merriam-webster.com/dictionary/methodology, accessed 03-May-2017.

[12] Ibrahim Haddad, Brian Warner, “Understanding the Open Source Development Model’’, A White Paper by The Linux Foundation, November 2011.

[13] Alessandra Cavarra, Jim Davies, Sergey Olvovsky, Alaen Harmet, “Using UML for Automatic Test Generation’’, Cavarra, 2001.

[14] Shailaja Ukea, Ravindra Thoolb, “UML Based Modeling for Data Aggregation in Secured Wireless Sensor Network’’, International Conference on Information Security & Privacy (ICISP2015), 11-12 December 2015, Nagpur, INDIA.

[15] Ceyhun Ozgur, Taylor Colliau, Grace Rogers, Zachariah Hughes, Elyse Myer-Tyson Valparaiso University, “MatLab vs. Python vs. R’’, Journal of Data Science 15(2017), 355-372. 2017.

[16] Huawei, “NodeB data configuration’’, available on https://www.slideshare.net/sheltonsiziba/140556299-huaweinode-bdataconfiguration-52749897’’, accessed on 05-11-2017.

[17] Alexandre Leclerc, “MATLAB to Python’’, Enthought United States, Version1.0, Sep-2017.

[18] Saba Irshad, Purna Chandra Nepal, “Rise over Thermal Estimation Algorithm Optimization and Implementation’’, Blekinge Institute of Technology February 2013.

[19] Defika Fiant, Uke Usma, Yuyun Siti Rohma “UMTS/HSPA Network KPI Value Optimization Analysis in UMTS State Transformation Feature Activation’’, The 3 rd International Conference on Radar, Antenna, Microwave, Electronics and Telecommunications (ICRAMET) 2014.

[20] Actix, “Cross feeder analysis’’ available on http://ftp.actix.com/resources/1502/Projects/KB/TechLayer/Crossed_feeder_Analysis.htm, accessed on 21-June-2017.

[21] Martin Lindberg, “Decision Support Diagnostics and Explanation Methods In the context of telecommunication networks’’, Umeå University, Umeå Sweden, April 2013.

[22] Martin Host, Alma Oru Alagi, “A Systematic Review of Research on Open Source Software in Commercial Software Product Development’’, Lund University Sweden.

[23] Vidyasagar Potdar, Elizabeth Chang, “Open Source and Closed Source Software Development Methodologies’’, School of Information System, Curtin University of Technology, Perth, Australia 6845, accessed on 15-8-2017.

[24] Ericsson, “Basic command moshel,’’ available on www.scribd.com/doc/172678125/ERICSSON-BASIC-COMMAND-MOSHEL-3G-RBS-6000-pdf’’, accessed on 12-July-2017.

[25] T. Azlinda, H. Amran, “Design of Tower Mounted Amplifier for WCDMA

Applications'', RF AND MICROWAVE CONFERENCE, OCTOBER 5 - 6, SUBANG, SELANGOR, MALAYSIA. 2004.

[26] Anristu, "Tower Mounted Amplifiers, "Diagnostics and Isolation Measurements'', available: https://dl.cdn-anritsu.com/en-us/test-measurement/files/Application-Notes/Application-Note/11410-00274.pdf, accessed on 16-11-2017.

[27] Huawei, "Radio Network Optimization Guide'', available on www.academia.edu/8473248/WCDMA_Radio_Network_Optimization_Guide_Table_of_Contents, accessed on 12-10-2017.

[28] Raquel Barco, Pedro Lazaro, "Automated fault Management in Wireless Networks'', Chapter XLVII, University of Malaga Spain, Handbook of Research on Telecommunications Planning and Management for Business.

[29] Mohammad Baig, "Radio Frequency Analysis of UMTS System'', Advance Networking and Telecom Dept (SOIT) International Institute of Information Technology Pune, India.

[30] Md. Al-Beruni, Mahfuz Alam, Rashedul Qayum, "Scope of Automated Organizing Network in Telecom Industry'', International Journal of Scientific & Engineering Research, Volume 4, Issue 9, September-2013 672, ISSN 2229-5518.

[31] Olav Østerbø , Ole Grøndalen, "Benefits of Self-Organizing Networks (SON) for Mobile Operators'', Journal of Computer Networks and CommunicationsVolume 2012 (2012), Article ID 862527.

[32] Ranjana Sridhar, "Self-Organizing Network'', ADET(P), "M"Division, TEC.

[33] Aftab A. Siddiqui, "Network Operations and Network Management'', available on https://www.sanog.org/resources/sanog17/sanog17-networkmgmt-aftab.pdf, accessed on 22-October-2017.

[34] K. Kumaravel, "Comparative Study of 3G and 4G in Mobile Technology'', IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 5, No 3, September 2011 ISSN (Online): 1694-0814.

[35] Sunil Patil, Y.S. Patil, "A review on outsourcing with a special reference to telecom operations'', Procedia - Social and Behavioral Sciences 133(2014) 400 – 416. 2014.

[36] Mihai DESPA, "Comparative study on software development methodologies'', Database Systems Journal vol. V, no. 3/2014.

[37] Claes Beckman, "Verifying 3G License Coverage Requirements'', Center for RF-Measurement Technology, University of Gävle, Sweden. 04-2011.

[38] E. Eter, Ericsson “UMTS UL Interference E. Eter’’, available on https://www.scribd.com/doc/190229011/UL-RSSI-Ericsson, accessed on 27-11-2017

[39] Osama. Bilal, Ericsson “UMTS Radio Network Tunning and Optimization’’, available on https://www.Ericsson.com, accessed on 29-01-2019

Appendix A: MATLAB CODES

A.1 Single-Site Mode Tool

```
function varargout = GUI(varargin)


% GUI MATLAB code for GUI.fig
%      GUI, by itself, creates a new GUI or raises the existing
%      singleton*.
%
%      H = GUI returns the handle to a new GUI or the handle to
%      the existing singleton*.
%
%      GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI.M with the given input arguments.
%
%      GUI('Property','Value',...) creates a new GUI or raises the
%      existing singleton*.  Starting from the left, property value pairs
are
%      applied to the GUI before GUI_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to GUI_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI

% Last Modified by GUIDE v2.5 23-Sep-2017 07:25:18

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @GUI_OpeningFcn, ...
    'gui_OutputFcn',  @GUI_OutputFcn, ...
    'gui_LayoutFcn',  [] , ...
    'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
```

```
% End initialization code - DO NOT EDIT


% --- Executes just before GUI is made visible.
function GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI (see VARARGIN)

% Choose default command line output for GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



function SiteID_Callback(hObject, eventdata, handles)
set (handles.txtmsg1,'String','');
set (handles.txtmsg2,'String','');
set (handles.txtmsg3,'String','');
set (handles.txtmsg4,'String','');
set (handles.txtmsg5,'String','');
set (handles.txtmsg6,'String','');
set (handles.txtmsg7,'String','');
set (handles.txtmsg8,'String','');
set (handles.txtmsg9,'String','');
% hObject    handle to SiteID (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of SiteID as text
%        str2double(get(hObject,'String')) returns contents of SiteID as a
double


% --- Executes during object creation, after setting all properties.
function SiteID_CreateFcn(hObject, eventdata, handles)

% hObject    handle to SiteID (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in CheckNumUsers.

function CheckNumUsers_Callback(hObject, eventdata, handles)
clc;


a=dir('*.txt');
j=1;nLog=[];
for i=1:size(a,1)
    fname=a(i).name;
    if  ~isempty(strfind(fname,'log')) && fname(4)~='.'
        nLog(j)=str2num(fname(4:length(fname)-4));
        j=j+1;
    end
end
if isempty(nLog)
    logFile=['log0.txt'];
else
    lastLog=max(nLog);
    logFile=['log' num2str(lastLog) '.txt'];
end








[IDandNUM SITE]=xlsread('users.xlsx');
 %%% Read Microsoft Excel File that contains users number
id=str2num(get(handles.SiteID,'String'));
setGlobal({num2str(id)},{logFile});
if (~isempty(id))
    i=find(IDandNUM(:,1)==id);
else
    i=0;
end

if (i>=1) && ~isempty(id)
    n_users=IDandNUM(i,3);
    set(handles.txtmsg1,'String',['The corresponding number of users is: '
num2str(n_users)]);
    if(n_users>90)
        set(handles.txtmsg2,'String','Please expand Hardware/Software
...');
    else
        set(handles.txtmsg2,'String','Please proceed with Troubleshooting
...');
    end
```

```
else
    set(handles.txtmsg1,'String','The Site ID is incorrect please try
again');
    set(handles.txtmsg2,'String','Please enter a valid Site ID ...');
end
wf=fopen(logFile,'at');
fprintf(wf,'\n');
fprintf(wf,[getTimeStr() ': Number of users checked for Site ID: '
num2str(id)]);
fclose(wf);
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
rf=fileread('Read File.txt');
i=strfind(rf,'STR CROSFEEDTST:');
if(strfind(lower(rf(i+20:i+30)),lower('NORMAL')))
    set(handles.txtmsg3,'String','Please proceed with Troubleshooting
...');
end
if(strfind(lower(rf(i+20:i+30)),lower('ABNORMAL')))
    set(handles.txtmsg3,'String','Please ask the FM Team to visit the site
...');
end
idlf=getGlobal;
logFile=cell2mat(idlf(2));
wf=fopen(logFile,'at');
fprintf(wf,'\n');

id1=cell2mat(idlf(1));
fprintf(wf,[getTimeStr() ': Crossfeeder checked for Site ID: ' (id1)]);
fclose(wf);
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
clc;
rf=fileread('Read File.txt');
i=strfind(rf,'List RRU/RFU Configure Information');
j=strfind(rf(1:i),'Number of results = ');
n_results=str2num(rf(j+20));

rf=textread('Read File.txt','%s','delimiter','\n');  %Read file line by
line

RXFix=0;
%RX Branch Check Result
a=strfind(rf,'List RxBranch Configure Information');
for k=1:length(a)
    if(cell2mat(a(k))==(1))
        break;
    end
end

commands=[];info=[];
for k=k+4:k+n_results+3
    matrf=cell2mat(rf(k));
    kk=1;
```

```
    while (length(info)<4)
        if ~isempty(str2num(matrf(kk)))
            ii=0;
            while ~isempty(str2num(matrf(kk+ii)))
                ii=ii+1;
            end

            info=[info str2num(matrf(kk:kk+ii))];

        end
        kk=kk+1;
    end

    if isempty(strfind(cell2mat(rf(k)),'ON')) ||
isempty([strfind(cell2mat(rf(k)),'16') strfind(cell2mat(rf(k)),'18')])
        commands=[commands;'MOD RXBRANCH:CN=' num2str(info(1)) ',SRN='
num2str(info(2)) ',SN=' num2str(info(3)) ',RXNO=' num2str(info(4))
',ATTEN=16;'];
        RXFix=1;
    end
    info=[];
end


idlf=getGlobal;
id1=cell2mat(idlf(1));
logFile=cell2mat(idlf(2));
wf=fopen(logFile,'at');
fprintf(wf,'\n');

fprintf(wf,[getTimeStr() ': RX Branch checked for Site ID: ' (id1)]);
fclose(wf);
%RRU Check Result

a=strfind(rf,'List RRU/RFU Configure Information');
for k=1:length(a)
    if(cell2mat(a(k))==(1))
        break;
    end
end
n_results2=0;
kk=k+5;
while isempty(strfind(lower(cell2mat(rf(kk))),lower('(Number of results')))
    n_results2=n_results2+1;
    kk=kk+1;
end

commands2=[];
for kk=k+5:k+n_results2+4
    info=[];
    matrf=cell2mat(rf(kk));
    kkk=1;
    while (kkk<length(matrf))&&length(info)<12

        if ~isempty(str2num(matrf(kkk)))
            ii=0;
            if (kkk+ii+1<=length(matrf))
                while ~isempty(str2num(matrf(kkk+ii+1)))
                    ii=ii+1;
```

```
                end
            end
            info=[info str2num(matrf(kkk:kkk+ii))];
        end
        kkk=kkk+ii+1;
        ii=0;
    end



    if (info(12)~=0)
        cm=['MOD RRU:CN=' (num2str(info(1))) ',SRN=' (num2str(info(2)))
',SN=' num2str(info(3)) ',RFDS=0;'];
        commands2=[commands2; cm];
    end

end

idlf=getGlobal;
id1=cell2mat(idlf(1));
logFile=cell2mat(idlf(2));

wf=fopen(logFile,'at');
fprintf(wf,'\n');

fprintf(wf,[getTimeStr() ': RRU checked for Site ID: ' (id1)]);
fclose(wf);
%TMA Result
a=strfind(rf,'Display TMA Subunit Dynamic Information');
for k=1:length(a)
    if(cell2mat(a(k))==(1))
        break;
    end
end

commands3=[];
for kk=k+4:k+n_results+3
    info=[];
    matrf=cell2mat(rf(kk));
    kkk=1;
    while (kkk<=length(matrf))

        if ~isempty(str2num(matrf(kkk)))
            ii=0;
            if (kkk+ii+1<=length(matrf))
                while ~isempty(str2num(matrf(kkk+ii+1)))
                    ii=ii+1;
                end
            end
            info=[info str2num(matrf(kkk:kkk+ii))];
        end
        kkk=kkk+ii+1;
        ii=0;
    end

    if info(4)~=48 || isempty(findstr(lower(matrf),lower('AVAILABLE'))) ||
isempty(findstr(lower(matrf),lower('NORMAL')))
        cm=['MOD TMASUBUNIT:DEVICENO=' (num2str(info(1))) ',SUBUNITNO='
(num2str(info(3))) ';'];
```

```
        commands3=[commands3; cm];
    end

end


wf=fopen('Write file.txt','wt');
for i=1:size(commands,1)
    fprintf(wf,(commands(i,:)));
    fprintf(wf,'\n');
end
for i=1:size(commands2,1)
    fprintf(wf,(commands2(i,:)));
    fprintf(wf,'\n');
end
for i=1:size(commands3,1)
    fprintf(wf,(commands3(i,:)));
    fprintf(wf,'\n');
end

fclose(wf);

if RXFix==0
    set(handles.txtmsg4,'String', 'RX Branch check is done');
else
    set(handles.txtmsg4,'String', 'RX Branch check is done. Command is sent
to fix problems.');
end
set(handles.txtmsg5,'String', 'RRU check is done. Command is sent to fix
problems.');
set(handles.txtmsg6,'String', 'TMA check is done. Command is sent to fix
problems.');

idlf=getGlobal;
id1=cell2mat(idlf(1));
logFile=cell2mat(idlf(2));

wf=fopen(logFile,'at');
fprintf(wf,'\n');

fprintf(wf,[getTimeStr() ': TMA checked for Site ID: ' (id1)]);
fclose(wf);




% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
wf=fopen('Write file.txt','at');
commands4='RST BRD:CN=0,SRN=4,SN=0;';
```

```
for i=1:size(commands4,1)
    fprintf(wf,(commands4(i,:)));
    fprintf(wf,'\n');
end

set(handles.txtmsg7,'String', 'Board reset is done');

fclose(wf);

idlf=getGlobal;
logFile=cell2mat(idlf(2));

wf=fopen(logFile,'at');
fprintf(wf,'\n');
fprintf(wf,[getTimeStr() ': Board reset is done.']);
fclose(wf);
```

```
% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)

wf=fopen('Write file.txt','at');
commands5='STR RFTEST:TSTTYPE=PIMONTRAFFIC_ONLINE;';
for i=1:size(commands5,1)
    fprintf(wf,(commands5(i,:)));
    fprintf(wf,'\n');
end

fclose(wf);


rf=fileread('Read File.txt');
i=strfind(rf,'STR RFTEST:TSTTYPE=PIMONTRAFFIC_ONLINE;');
if(strfind(lower(rf(i+40:i+50)),lower('NORMAL')))
    set(handles.txtmsg8,'String','Interference is Normal');
    set(handles.txtmsg9,'String','');

end
if(strfind(lower(rf(i+40:i+50)),lower('ABNORMAL')))
    set(handles.txtmsg8,'String','Interference is Abnormal.');
    set(handles.txtmsg9,'String','Please Send the RF Team to visit the site
...');

end


idlf=getGlobal;
logFile=cell2mat(idlf(2));


wf=fopen(logFile,'at');
```

```
fprintf(wf,'\n');
fprintf(wf,[getTimeStr() ': Interference is checked.']);
fclose(wf);


% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)








a=dir('*.txt');
j=1;nLog=[];
for i=1:size(a,1)
    fname=a(i).name;
    if ~isempty(strfind(fname,'log')) && fname(4)~='.'
        nLog(j)=str2num(fname(4:length(fname)-4));
        j=j+1;
    end
end
if isempty(nLog)
    logFile=['log0.txt'];
else
    lastLog=max(nLog);
    logFile=['log' num2str(lastLog+1) '.txt'];
end
asd=fopen(logFile,'wt');
fclose(asd);

Dlf=getGlobal;
if isempty(Dlf)
    setGlobal({[]},{logFile});
else

    lf=(logFile);
    id=cell2mat(Dlf(1));


    setGlobal({id},{(lf)});

end

 %% Operation log

function dt=getTimeStr()
cl=clock;
if (cl(4)>9)
```

```
h=num2str(cl(4));
else
  h=['0' num2str(cl(4))];
end

if (cl(5)>9)
m=num2str(cl(5));
else
  m=['0' num2str(cl(5))];
end

if (round(cl(6))>9)
s=num2str(round(cl(6)));
else
  s=['0' num2str(round(cl(6)))];
end
t=[h ':' m ':' s];



if (cl(1)>9)
y=num2str(cl(1));
else
  y=['0' num2str(cl(1))];
end

if (cl(2)>9)
mn=num2str(cl(2));
else
  mn=['0' num2str(cl(2))];
end

if (round(cl(3))>9)
d=num2str(round(cl(3)));
else
  d=['0' num2str(round(cl(3)))];
end

date=['[' d '/' mn '/' y ']'];
dt = [date ' ' t];
end

function r=getGlobal
global x
r=x;
end
```

A.2 Multi-Site Mode Tool

```
function varargout = GUI(varargin)
clc

% GUI MATLAB code for GUI.fig
%      GUI, by itself, creates a new GUI or raises the existing
%      singleton*.
%
%      H = GUI returns the handle to a new GUI or the handle to
```

```
%       the existing singleton*.
%
%       GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%       function named CALLBACK in GUI.M with the given input arguments.
%
%       GUI('Property','Value',...) creates a new GUI or raises the
%       existing singleton*.  Starting from the left, property value pairs
are
%       applied to the GUI before GUI_OpeningFcn gets called.  An
%       unrecognized property name or invalid value makes property
application
%       stop.  All inputs are passed to GUI_OpeningFcn via varargin.
%
%       *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows only one
%       instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI

% Last Modified by GUIDE v2.5 22-Sep-2017 17:59:07

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
    'gui_Singleton',  gui_Singleton, ...
    'gui_OpeningFcn', @GUI_OpeningFcn, ...
    'gui_OutputFcn',  @GUI_OutputFcn, ...
    'gui_LayoutFcn',  [] , ...
    'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI is made visible.
function GUI_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI (see VARARGIN)

% Choose default command line output for GUI
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```

```
% --- Outputs from this function are returned to the command line.
function varargout = GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --------------------------------------------------------------------
function Import_Callback(hObject, eventdata, handles)


a=dir('*.txt');
j=1;nLog=[];
for i=1:size(a,1)
    fname=a(i).name;
    if  ~isempty(strfind(fname,'log')) && fname(4)~='.'
        nLog(j)=str2num(fname(4:length(fname)-4));
        j=j+1;
    end
end
if isempty(nLog)
    logFile=['log0.txt'];
else
    lastLog=max(nLog);
    logFile=['log' num2str(lastLog) '.txt'];
end


% hObject    handle to Import (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename filepath]=uigetfile('*.txt','Please select data
files','MultiSelect','On');
[num txt]=xlsread('Site list.xlsx');

s=length(filename);
if ~iscell(filename)
    s=1;
end

 %%% determine whether the input is call array

for i=1:s
    if iscell(filename)
        a=cell2mat(filename(i));
    else
        a=filename;
    end
    No(i)=i;
    siteID(i)=str2num(a(1:(strfind(a,'.txt'))-1));
    b=find(num(:,2)==siteID(i));
    if(b>0)
        nUsers(i)=num(b,3);
    else
```

```
        nUsers(i)=-1;
    end
    if (nUsers(i)>90)
        notes(i,:)=('Number of users exceeded 90, Software and hardware 
Check is required.');
    elseif (nUsers(i)==-1)
        notes(i,:)=('Missing site information: Number of users is not 
found.                  ');
    else
        notes(i,:)=('
');
    end
end
Data=[num2cell(No)' num2cell(siteID)',num2cell(nUsers)',cellstr(notes)];

lf=logFile;
for k=1:size(Data,1)-1
    lf=[lf;lf(1,:)];
end
%
setGlobal(Data,cellstr(lf));

set(handles.table,'data',Data);
set(handles.label1,'String','Number of users is checked please proceed with 
troubleshooting...');
set(handles.text2,'String','');
set(handles.text3,'String','');
set(handles.text4,'String','');
set(handles.text5,'String','');

wf=fopen(lf(1,:),'at');
fprintf(wf,'\n');
fprintf(wf,[getTimeStr() ': Site ID(s): ']);
for i=1:size(Data,1)-1
    fprintf(wf,[num2str(siteID(i)) ', ']);
end
if isempty(i)
    i=0;
end

fprintf(wf,[num2str(siteID(i+1)) ' has/have been imported for 
troubleshooting. ']);
fprintf(wf,'\n');
fclose(wf);
% --- Executes on button press in checkInterference.
function checkInterference_Callback(hObject, eventdata, handles)
% hObject    handle to checkInterference (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
wf=fopen('Write file.txt','wt');

commands=[];

[Dlf]=getGlobal;
lf=cell2mat(Dlf(end));
Data=(Dlf(:,1:end-1));

a=Data;

for j=1:size(getGlobal,1)
```

```
    cm=['STR RFTEST [' num2str(cell2mat(Data(j,2))) '];'];
    fprintf(wf,cm);
    fprintf(wf,'\n');



    rf=fileread([num2str(cell2mat(a(j,2))) '.txt']);
    i=strfind(rf,'STR RFTEST:TSTTYPE=PIMONTRAFFIC_ONLINE');
    if(strfind(lower(rf(i+40:i+50)),lower('NORMAL')))
        a(j,5)=cellstr('NORMAL');
    end
    if(strfind(lower(rf(i+40:i+50)),lower('ABNORMAL')))
        a(j,5)=cellstr('ABNORMAL');
    end

end
set(handles.table,'data',a);
set(handles.text2,'String','Interference is checked, please proceed with
troubleshooting...');

wf=fopen(lf,'at');
fprintf(wf,'\n');
fprintf(wf,[getTimeStr() ': Interference is checked for Site ID(s) ']);
for i=1:size(Data,1)-1
    fprintf(wf,[num2str(cell2mat(Data(i,2))) ', ']);
end
if isempty(i)
    i=0;
end

fprintf(wf,[num2str(cell2mat(Data(i+1,2))) '. ']);
fprintf(wf,'\n');

fclose(wf);
% --- Executes on button press in sendmail.
function sendmail_Callback(hObject, eventdata, handles)
[Dlf]=getGlobal;
lf=cell2mat(Dlf(end));
Data=(Dlf(:,1:end-1));

a=Data;

sids=[];
for i=1:size(Data,1)
    sids=[sids ' ' num2str(cell2mat(Data(i,2)))];
end


UserName='mohammedhisham375@gmail.com';
passWord='wnkfjfbzeodgoyji';
setpref('Internet','E_mail',UserName);
setpref('Internet','SMTP_Server','smtp.gmail.com');
setpref('Internet','SMTP_Username',UserName);
setpref('Internet','SMTP_Password',passWord);
props = java.lang.System.getProperties;
props.setProperty('mail.smtp.auth','true');
props.setProperty('mail.smtp.socketFactory.class', ...
    'javax.net.ssl.SSLSocketFactory');
```

```
props.setProperty('mail.smtp.socketFactory.port','465');

emailto = 'hisham_tcom@hotmail.com'; % recipient's email

mailtext=['Please check the following sites with interference issues:' sids
'.'];
sendmail(emailto, 'RTWP', mailtext);


set(handles.text5,'String','Notification email is sent to RF Team, please
proceed with troubleshooting... ');
wf=fopen(lf,'at');
fprintf(wf,'\n');
fprintf(wf,[getTimeStr() ': Notification email is sent to RF Team regarding
Site ID(s): ']);
for i=1:size(Data,1)-1
    fprintf(wf,[num2str(cell2mat(Data(i,2))) ', ']);
end
if isempty(i)
    i=0;
end

fprintf(wf,[num2str(cell2mat(Data(i+1,2))) '. ']);
fprintf(wf,'\n');

fclose(wf);
% hObject    handle to sendmail (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% function setGlobal(val,val1)
% global x;
% global x1;
% [x,x1]=deal(val, val1);
%
%
% function r=getGlobal
% global x;
% global x1;
% r=x, x1;

% --- Executes on button press in increaseTX.
function increaseTX_Callback(hObject, eventdata, handles)
wf=fopen('Write file.txt','at');
[Dlf]=getGlobal;
lf=cell2mat(Dlf(end));
Data=(Dlf(:,1:end-1));

a=Data;

for j=1:size(getGlobal,1)

    cmd1=['ADD CELLSIMULOAD: LocalCellId=' num2str(cell2mat(Data(j,2))) '5,
SimLoadCfgIndex=9,[' num2str(cell2mat(Data(j,2))) '];'];
    cmd2=['ADD CELLSIMULOAD: LocalCellId=' num2str(cell2mat(Data(j,2))) '6,
SimLoadCfgIndex=9,[' num2str(cell2mat(Data(j,2))) '];'];
    cmd3=['ADD CELLSIMULOAD: LocalCellId=' num2str(cell2mat(Data(j,2))) '7,
SimLoadCfgIndex=9,[' num2str(cell2mat(Data(j,2))) '];'];

    fprintf(wf,cmd1);
```

```
    fprintf(wf,'\n');
    fprintf(wf,cmd2);
    fprintf(wf,'\n');
    fprintf(wf,cmd3);
    fprintf(wf,'\n');

end

set(handles.text3,'String','Power is increased, please proceed with
troubleshooting...');
wf=fopen(lf,'at');
fprintf(wf,'\n');
fprintf(wf,[getTimeStr() ': Power is increased for Site ID(s): ']);
for i=1:size(Data,1)-1
    fprintf(wf,[num2str(cell2mat(Data(i,2))) ', ']);
end
if isempty(i)
    i=0;
end

fprintf(wf,[num2str(cell2mat(Data(i+1,2))) '. ']);
fprintf(wf,'\n');
fclose(wf);

% hObject    handle to increaseTX (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in BlockCells.
function BlockCells_Callback(hObject, eventdata, handles)
wf=fopen('Write file.txt','at');
[Dlf]=getGlobal;
lf=cell2mat(Dlf(end));
Data=(Dlf(:,1:end-1));

a=Data;

for j=1:size(getGlobal,1)

    cmd1=['BLK CELL: CellID=' num2str(cell2mat(Data(j,2))) '5, ['
num2str(cell2mat(Data(j,2))) '];'];
    cmd2=['BLK CELL: CellID=' num2str(cell2mat(Data(j,2))) '6, ['
num2str(cell2mat(Data(j,2))) '];'];
    cmd3=['BLK CELL: CellID=' num2str(cell2mat(Data(j,2))) '7, ['
num2str(cell2mat(Data(j,2))) '];'];

    fprintf(wf,cmd1);
    fprintf(wf,'\n');
    fprintf(wf,cmd2);
    fprintf(wf,'\n');
    fprintf(wf,cmd3);
    fprintf(wf,'\n');

end

set(handles.text4,'String','Cells are blocked.');
wf=fopen(lf,'at');
fprintf(wf,'\n');
```

```
fprintf(wf,[getTimeStr() ': Cells are blocked for Site ID(s): ']);
for i=1:size(Data,1)-1
    fprintf(wf,[num2str(cell2mat(Data(i,2))) ', ']);
end
if isempty(i)
    i=0;
end

fprintf(wf,[num2str(cell2mat(Data(i+1,2))) '. ']);
fprintf(wf,'\n');
fclose(wf);
% hObject    handle to BlockCells (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)



% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
a=dir('*.txt');
j=1;nLog=[];
for i=1:size(a,1)
    fname=a(i).name;
    if ~isempty(strfind(fname,'log')) && fname(4)~='.'
        nLog(j)=str2num(fname(4:length(fname)-4));
        j=j+1;
    end
end
if isempty(nLog)
    logFile=['log0.txt'];
else
    lastLog=max(nLog);
    logFile=['log' num2str(lastLog+1) '.txt'];
end
asd=fopen(logFile,'wt');
fclose(asd);

[Dlf]=getGlobal;
if isempty(Dlf)
    setGlobal({[]},{logFile});
else

    lf=(logFile);
    Data=(Dlf(:,1:end-1));

    a=Data;
    for k=1:size(Data,1)-1
        lf=[lf;lf(1,:)];
    end

    setGlobal(a,cellstr(lf));

end


%% Operation log
```

```matlab
function dt=getTimeStr()
% Get time string in HH:MM:SS
cl=clock;
if (cl(4)>9)
h=num2str(cl(4));
else
  h=['0' num2str(cl(4))];
end

if (cl(5)>9)
m=num2str(cl(5));
else
  m=['0' num2str(cl(5))];
end

if (round(cl(6))>9)
s=num2str(round(cl(6)));
else
  s=['0' num2str(round(cl(6)))];
end
t=[h ':' m ':' s];


if (cl(1)>9)
y=num2str(cl(1));
else
  y=['0' num2str(cl(1))];
end

if (cl(2)>9)
mn=num2str(cl(2));
else
  mn=['0' num2str(cl(2))];
end

if (round(cl(3))>9)
d=num2str(round(cl(3)));
else
  d=['0' num2str(round(cl(3)))];
end

date=['[' d '/' mn '/' y ']'];
dt = [date ' ' t];
end

function r=getGlobal
global x
r=x;
end
```