Master's thesis

Software Engineering and ICT

2024

Perttu Pisilä

Data Representation Requirements on Network Measurement Applications

TURKU AMK

TURKU UNIVERSITY OF APPLIED SCIENCES

Master’s Thesis | Abstract

Turku University of Applied Sciences

Master of Engineering, Software Engineering and ICT

2024 | 87 pages

Perttu Pisilä

Data Representation Requirements on Network Measurement Applications

Data representation is an essential part of a mobile network measurement application. The objective of this thesis is to elicit and analyze data representation related user requirements for network measurement applications, based on existing functionality of network measurement applications Nemo Outdoor and Nemo Handy-A.

Mixed-method approach is used to form an understanding of the type of data being recorded by these applications and their current data representation capabilities. New data representation requirements are elicited using interviews as the data collection and requirements elicitation method.

The main finding is a need for use case specific data view configurations. As the network measurement applications are utilized in various use cases from troubleshooting to benchmarking, the user interface shall adapt to the use case by representing the most relevant data in a user-friendly form.

Keywords:

Data Representation, Requirements Engineering, Mobile Networks

Opinnäytetyö (YAMK) | Tiivistelmä

Turun ammattikorkeakoulu

Insinööri (ylempi AMK), ohjelmistotekniikka ja ICT

2024 | 87 sivua

Perttu Pisilä

Datan esittämisen vaatimukset matkapuhelinverkkojen mittaussovelluksissa

Datan esittäminen on keskeisessä osassa matkapuhelinverkkojen mittaussovelluksissa. Tämän opinnäytetyön tavoitteena on kerätä ja analysoida datan esittämiseen liittyviä vaatimuksia matkapuhelinverkkojen mittasovelluksissa, pohjautuen mittaussovellusten Nemo Outdoor ja Nemo Handy-A nykyiseen toiminnallisuuteen.

Sovellusten tuottamaa dataa ja niiden tämänhetkisiä datan esityskyvykkyyksiä pyritään ymmärtämään monimenetelmällisen tutkimuksen avulla. Uusia vaatimuksia datan esittämiselle sovelluksissa kartoitetaan käyttämällä haastatteluja tiedonkeruu- ja vaatimusten kartoitusmenetelmänä.

Työn keskeisin löydös on tarve käyttötapauskohtaisille datanäkymien konfiguraatioille. Koska langattomien verkkojen mittaussovelluksia käytetään useisiin eri tarpeisiin vian etsinnästä eri verkkojen vertailuun, tulisi niiden käyttöliittymien sopeutua käyttötapaukseen esittämällä siihen olennaisesti kuuluva data käyttäjäystävällisessä muodossa.

Asiasanat:

Datan esittäminen, vaatimusmäärittely, matkapuhelinverkot

Content

Appendices

Figures

Tables

List of abbreviations

3GPP	The 3rd Generation Partnership Project. A group of standard development organizations producing specifications for mobile network technologies [1].
BTS	Base Transreceiver Station. “A piece of network equipment that facilitates wireless communication between a device and network.” [2]
DUT	Device Under Test. Typically a smartphone or a network scanner in the context of this study.
IoT	Internet of Things. A global infrastructure of interconnected physical or virtual objects [3].
KA	Knowledge Area. A major category of engineering topics in the SWEBOK [4].
KML	Keyhole Markup Language, currently known simply as KML. File format for geographic data [5].
KPI	Key Performance Indicator. A strategically selected measurable value of the performance of a system.
MOS	Mean Opinion Score. Used to assess the voice quality of a voice call [6].
NATA	Nemo Active Testing Application. A smartphone application and a communication interface used with PC based network measurement tools.
RSRP	Reference Signal Received Power. A signal level parameter that can be measured from LTE and NR networks [7], [8].

RSRQ	Reference Signal Received Quality. A signal quality parameter that can be measured from LTE and NR networks [7], [8].
SWEBOK	Software Engineering Body of Knowledge. A collection of generally accepted knowledge of software engineering, also recognized as ISO Technical Report 19759 [4].
UML	Unified Modeling Language. A modelling language designed to specify, visualize, and document models of software systems [9].
UI	User Interface. The part of the software, the user is interacting with.
UX	User experience. Users overall experience interacting with products, such as a software application.
XML	Extensible Markup Language. Text format specification, typically stored in files with .xml extension [10]. Used in display configuration files for Nemo tools.

1 Introduction

The objective for the thesis is to elicit and analyze user requirements for data representation on network measurement applications.

Mobile networks and access to mobile broadband has become essential part of people's lives in the 21st century. In developing countries, mobile networks provide the primary way to access the Internet for most of the population [11]. It is even argued that Internet access should be recognized as a human right, based on its instrumental value on other human rights, including social security, education, health, housing and work [12].

In OECD countries, monthly mobile data usage reached 10.4 GB per subscription in 2022, doubling in four years [13]. Rollout of fifth generation networks has gathered pace and estimated 1.6 billion 5G subscriptions worldwide in 2023 is forecasted to exceed to 5.3 billion in 2029 [14].

1.1 Background and Motivation

To ensure this critical infrastructure is performing optimally and new technologies can be developed and deployed, it is essential to have tools to measure and analyze the mobile networks.

The network measurement applications, also called collection tools, typically have multiple functions: configure the DUT (device under test), execute tests on the DUT, collect data from the DUT and represent the collected data. Data collected from the DUT is typically saved into a log file for later analysis. The data is also often represented in real time on the application’s user interface (UI) during the measurement using numerical and textual values or with more advanced visualization techniques like graphs and maps. Some measurement tools also have a capability to playback previously recorded log files and create reports.

The log files with the network measurement data can be post-processed in a variety of analytics software and converted into reports. These reports can then be used to help in the development and maintenance of the networks and network devices. There are many use cases, however, where this type of process is too heavy and time intensive. A network issue might be troubleshooted by an expert in the field, given that the relevant data is accessible in the measurement tool and represented in a meaningful way.

Data representation and visualization have a crucial role in bringing out the value that is embedded in the collected data. They offer the user a way to understand the data and find patterns from it.

One of the companies competing in the network testing field is Keysight Technologies with their Nemo Wireless Network Solutions [15], hereafter referred to as “Nemo”, which is introduced in more detail in chapter 4. This research is commissioned by Keysight Technologies Finland Oy, a Finnish limited company that is part of an American technology company, Keysight Technologies. This study is based around their network measurement products Nemo Outdoor and Nemo Handy-A.

Both Nemo Outdoor and Nemo Handy-A are established software in the network measurement market and under continuous development. Still, many parts of their core functionality and architecture are based on the original design, while the operating environment is radically different from the time when these software applications were first written. They also share a lot of the same functionality, while they often have unique implementations.

To address these issues, enhancement projects has been planned to modernize multiple areas of the architecture and find synergies by sharing more code between the two products. One of these areas is the data architecture: how the measurement data is stored and represented. This study is the first step towards this goal by eliciting and analysing the user requirements for the data representation.

The author of this thesis has worked for the commissioner for ten years in the roles of Trainee, Customer Support Engineer and Software Developer, working with many of the stakeholders that are relevant for this research. The author has provided hands-on training for Nemo collection tools to the end users and developed features and fixes for the Nemo Handy-A product.

1.2 Literature Review

Literature review covered existing research, theses, books, specifications and other literature related to the topics of this thesis.

1.2.1 Mobile Network Measurements

The 3rd Generation Partnership Project (3GPP) provides technical specifications for measuring LTE [7], NR [8] and other mobile network technologies. Goel et al. [16] surveyed mobile network measurement testbeds, tools, and services for end-to-end testing. The survey covered 20 different solutions, focusing on their functionality in assessing mobile network performance against the needs of application developers, researchers, network operators, and regulators. The survey did not cover the data representation of the solutions, nor did it cover any Nemo tools.

Public research of Nemo collection tools is limited mostly on using them to solve a particular network problem. An example of such research is Sanaullah's master's thesis on the narrowband internet of things (NB-IoT) network coverage measurements.

1.2.2 Data Visualization

Jiang et al. [17] conducted a literature review on the development of visualization studies that found 1930 articles related to data visualization and additional 299 articles on information visualization published between 2012 and

2022. The number of published articles on data visualization is increasing year by year. The study defines scientific visualization as “a further extension of data visualization to the real perceptual image” and finds information visualization more inclined towards visual communication. The study finds high versatility on the topmost co-cited articles with a high degree of inter-article correlation.

A visualization study that best matched with the themes of this research was Tiirikainen’s [18] master’s thesis about visualization of log data. Although the thesis is grounded in visualization theory, including how knowledge is derived from the data, it focuses on the implementation of the visualization system for the real time wireless network measurement data, instead of the user perspective on the value of that data.

1.2.3 Software Requirements

“Software requirements” is the first of the fifteen knowledge areas (KAs) recognized by the Guide to the Software Engineering Body of Knowledge (SWEBOK) [4]. Based on existing literature, it presents the established requirements engineering practices in the field of software engineering.

Requirements engineering has been the theoretical framework for multiple master’s theses, including Vainio’s [19] partially confidential work about customer-oriented collaboration in requirements engineering, and Jussila’s [20] work about the user interface as a part of the requirements specification. While the results of these work don’t provide the customer requirements for network measurement applications, the approach of using requirements engineering as the theoretical framework is applicable and used in this thesis as well.

1.2.4 Research Gap

This thesis is positioned at the intersection of mobile network measurements, data representation and requirements engineering, where the research gap exists. The literature reviewed for this thesis couldn’t provide the user requirements for the data representation that could be applied to the Nemo collection tools directly, so they are developed using requirement engineering techniques.

1.3 Research Questions

The objective for the research in this thesis is to elicit and analyze user requirements for data representation on network measurement applications, specifically for Nemo Outdoor and Nemo Handy-A. The main research question is: What are the data representation related user requirements on network measurement tools?

To help answer the main research question, three sub-questions are defined:

1. Where is the value in network measurement data for the users?
2. What data representation limitations do Nemo collection tools have?
3. What type and quantity of data do network measurement tools record in Nemo File Format?

1.4 Outline of the Thesis

The outline of this report follows the IMRaD format of introduction, methods, results and discussion, with slight adjustments. Following the general introduction and background of the work in chapter 1, the methodology of this research is introduced in chapter 2. It covers each of the methods used and their limitations. The methodology is followed by the theoretical framework in chapter 3, presenting requirements engineering principles and how they are applied on this study.

The results are presented in the following three chapters for each of the three phases of the study. First, a background of the Nemo collection tools is provided with the presentation of the product’s current data representation capabilities in chapter 4 and Appendix 1. Then the properties of network measurement data collected and presented by Nemo collection tools are covered in the chapter 5. Finally, the user requirements for data representation are analyzed in chapter 6. The results are summarized in chapter 7 by answering the research questions.

The work is summarized, and future work is discussed in the Conclusion chapter at the end of the report.

2 Methodology

This study uses a mixed-method research approach with comparative, quantitative and qualitative components. Each component is part of a requirements engineering process as described in theory section 3.4.

2.1 Comparative Research

The comparative component of this study is designed to work as an introduction to the data representation capabilities of established network measurement tools Nemo Outdoor and Nemo Handy-A. The data representation capabilities of both products are analyzed separately on sections 4.3 and 4.4, respectively. Comparison tables of each type of data views are compiled in Appendix 1. These capabilities are used as a baseline for the new requirements being elicited and analyzed in the quantitative component of this study (see section 2.3)

2.1.1 Data Collection Methods

The data for the comparative research was collected by analyzing the user interface of each product, their documentation and source code. In few instances, when the functionality was unclear, product's developers were consulted for clarification. Product versions used for the data collection were Nemo Outdoor 9.60 and Nemo Handy-A 4.90. All screenshots of these software included in this report are captured by the author.

2.1.2 Limitations

The author is more familiar with Nemo Handy-A product, both its UI and source code, compared to Nemo Outdoor. Most of the UI analysis was done using the playback functionality, where some of the UI features differ from a real time measurement. Both products adjust the available features based on the used license and available data. The licenses used for the analysis had all the necessary feature options enabled. Multiple measurement files were used to ensure that all data views were covered in the analysis. Source code and documentation analysis were also incorporated to cover blind spots of the UI analysis.

The two products are used on devices with different form factors and have unique architecture and implementations. Therefore, direct comparison of the data views is not possible. The comparison tables in the Appendix 1 are based more on the terminology and features available in Nemo Outdoor. They are designed to cover all the main features, but all the configuration options might not be listed in the tables.

2.2 Quantitative Research

The quantitative component of this study is designed to provide insight into the type and quality of the network measurement data recorded by the Nemo collection tools and provide answers to the third sub research question: “What type and quantity of data do network measurement tools record in Nemo File Format?”

Nemo File Format is an event-based file format for the network measurement log files, written by Nemo Outdoor, Nemo Handy-A and various other collection tools. A log file in Nemo File Format starts with header events, followed by the measurement events and footer events. The events may be encrypted or plain text in comma separated form.

The insight of the real-world data quantity is crucial when requirements for the data architecture and visualization components are developed. The analysis of the data is presented in chapter 5. The quantitative research is based on an exercise the author conducted preceding this thesis as a part of their master’s degree [21].

2.2.1 Data Collection Methods

The population of interest comprises network measurement files produced by Nemo collection tools. Vast majority of this population is inaccessible as it is proprietary to the companies conducting the measurements. Sampling frame used in this research comprises of the measurements conducted internally by the Nemo organization. As the network technologies and collection tools have evolved, measurements over five years old were filtered out. The selected sample comprises of measurements conducted by the Nemo testing team between January 2018 and December 2023.

The sample is processed with a program written in Kotlin. The program recursively iterates over the measurement directories. It extracts all compressed measurement files and decrypts encrypted measurement files into unencrypted format. The measurement file is parsed into two SQLite database tables. In the “measurement” table, each measurement file is given a sequential identifier number, and the values of all the header and footer events are parsed to corresponding columns. The “measurement” table also has a column for the full path of the measurement file. The program calculates metrics of each unique measurement event type on the file, including total number of events, minimum time between events of the that type, and total number of parameters (commas in the events of that type). Minimum, maximum, mean and mode values for the parameters within an event are also calculated. This information is written to a “event_info” table with the measurement identifier number that allows joining it with the “measurement” table.

The SQL database was then imported and further processed with an R script with following steps:

1. Import the SQLite tables (measurement, event_info) as data frames.
2. Remove all entries from *measurement*, that don’t have any events in the *event_info*.
3. Remove entries from both data frames, if the measurement is missing #ID header or #STOP footer.

4. Remove duplicate entries from both data frames, identified by the combination of #EI and #ID values.
5. Convert measurement start and stop times into POSIXct format.
6. Calculate measurement duration into a “duration” column in the *measurement* data frame.
7. Remove entries from both data frames, if the duration is invalid, under a minute or over 5 hours.
8. Calculate total number of events and parameters in each measurement as new columns in the *measurement* data frame, based on the information in the *event_info* data frame.
9. Calculate event density and parameter density for each measurement as new columns in the *measurement* data frame.
10. Calculate event density and parameter density for each event type within the measurement as new columns in the *event_info* table.
11. Parse the product name and version number as new columns in the *measurement* data frame, based on the #PRODUCT header.
12. Parse the NATA and version number as new column in the *measurement* data frame, based on the #NMR header.
13. Cleanup #FF and #DT header values.
14. Remove entries from both data frames if the measurement is done with Nemo Outdoor without using NATA (too few measurements in the sample).
15. Remove entries from both data frames if the measurement is identified as containing invalid data through manual inspection.

2.2.2 Material

After the data processing (described above), the dataset contains 5691 measurements, of which 138 are from a network scanner. The DUT type in a vast majority of the measurement in the dataset is a phone. Distribution of the measurements between products and DUT types is presented in Table 1.

Table 1. Number of measurements in the dataset.

Product	Phone	Scanner	Total
Nemo Handy-A	3080	105	3185
Nemo Outdoor	2473	33	2506
Total	5553	138	5691

The dataset contains total of 43 unique header or footer events and 183 unique measurements events. The dataset contains 175 and 19 unique measurement events for phone and scanner measurements, respectively.

2.2.3 Limitations

The sample selected for this research is produced by the Nemo testing team and contains measurement files produced by pre-release software. These pre-release builds may contain defects or logic that differ from the released software and distorts the data. The test cases, on which the measurement files in the sample are based, are designed for validating the software and might not resemble the real-world use cases of the end customers.

The analysis of the measurement files was done purely on their comma separated form, without referring to the file format specification. Parsing the parameters based on the file format could result to more accurate outcome.

2.3 Qualitative Research

The qualitative component of this study is designed to answer the main research question and its sub questions about the data representation requirements of network measurement tools. By answering the sub research questions: “Where is the value in network measurement data for the users?” and “What data representation limitations do Nemo collection tools have?”, user requirements for a network measurement tools can be developed, which in turn answer the main research question: “What are the data representation related user requirements on network measurement tools?”

The analysis of the quantitative research component is presented in chapter 6.

2.3.1 Data Collection Methods

The original plan for the qualitative component of this study was to use observations as the data collection method and requirements elicitation technique. Due to reasons described in the Limitations section below, the technique was changed to interviews.

Total of seven interviews were conducted between September 6th 2023 and November 14th 2023, including ten individual stakeholders of Nemo collection software. First five of the interviews were one-on-one interviews with experts within the Nemo organization that have worked on multiple roles within the organization, including customer support, sales support, product management and other management and specialist roles. These stakeholders are labeled with “Nemo_” prefix in the analysis section and Table 2.

The two customer facing interviews were small-group interviews with two and three participants from each company, respectively. The first company provides consulting and subcontracting services related to mobile networks, and the second company is a mobile network operator. These stakeholders are labeled with “Consultant_” and “Operator_” prefixes, respectively. Both companies have experience in using Nemo collection and post processing tools.

The first interview was held face-to-face, and the other six interviews were conducted remotely via Microsoft Teams. The interviews were semi-structured, covering various themes that were selected to match the expertise of the interviewees to support the requirements elicitation. Discussed themes of each interview are listed in Table 2.

Table 2. Interview themes.

Interviewees	Themes
Nemo_1	Use cases, display configurations, Nemo Handy-A serving info, scanner measurements
Nemo_2	Requirements engineering, stakeholders, use cases, automation
Nemo_3	Data views, use cases, user requirements
Nemo_4	North American market, use cases, Nemo Handy-A usability
Nemo_5	Nemo Handy-A data views, usability and configurability
Consultant_1, Consultant_2	Setup and use cases, KPIs, post processing, limitations of Nemo tools, notifications
Operator_1, Operator_2, Operator_3	Setup and use cases, KPIs, Nemo Outdoor data views, limitations of Nemo tools

The interviews were supported by a simple set of slides that contained information about the author, research questions, major use cases for network measurement tools, and screenshot of Nemo collection tools. Minor adjustments for the slides were made for some of the interviews. The supporting slides are provided in Appendix 2 in their general form with the original research questions. The additional research question, as described in limitations section 2.3.3, is omitted from the appendix. In their interview, Nemo_4 presented their ideas using screen sharing of Nemo Handy-A. In the second customer interview, the author used screen sharing of Nemo Outdoor to support the interview.

2.3.2 Material

The first interview that was held face-to-face was not recorded, but the author wrote notes totaling 749 words. The other six interviews were recorded on video totaling 5 hours and 34 minutes. The Finnish interviews were transcribed manually by the author, using intelligent verbatim transcription with timestamps that help reviewing to the original recording when needed. The English interviews were transcribed automatically by Microsoft Teams. The author manually cleaned the automated transcriptions to match the style of the other transcriptions and adjust the interviewee names on the second customer interview, where the two people shared the same Teams profile.

The transcribed material totaled 37800 words, of which 64% were said by the interviewees. Notes were then written under 16 themes, totaling 1883 words on 200 notes. During the analysis (see chapter 6), each of the 47 citations were marked on the transcription with the section number of the report. To prevent disclosure of sensitive or proprietary information of the customers or the commissioner, the transcriptions are not provided as appendices.

2.3.3 Limitations

Major changes occurred in the Nemo organization during the preparation phase of the qualitative research. Due to these changes, the plans for implementing the research had to be changed and the original plan of using observations as the primary research technique was rejected. The material used on this research is based on the interviews that was originally intended for preparing the sessions where the author could observe the end users using the Nemo collection tools. Portions of each interview was also used to discuss additional research question related to a possible upcoming feature, but this additional research question was dropped as the plan for the research was adjusted.

2.4 Usage of AI

OpenAI’s ChatGPT [22] with models GPT-3.5 and GPT-4 is used as a writing coach for this report. ChatGPT is used to help with word choices, connotation and nuance, as the author is not a native English speaker. ChatGPT was also utilized when writing the R scripts for the data manipulation of the quantitative research component. Prompts on the platform are written in a way that do not disclose the details of the research.

ChatGPT or other AI tools are not used to generate content for this report or as a source of facts. ChatGPT or other AI tools are not used to process any of the data or other material related to this study.

3 Requirements Engineering

The systematic handling of requirements is often referred to as “requirements engineering” and the management of those requirements during the life cycle of the product as “requirements management” [4]. The main requirements engineering activities of elicitation, analysis, specification and validation are grouped under “requirements development” by Wiegers and Beatty [23] as shown on Figure 1.

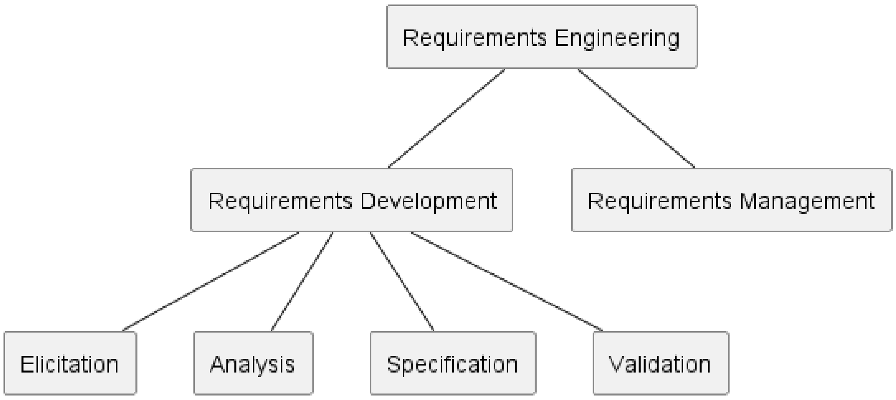


Figure 1. Subdisciplines of software requirements engineering [23].

This study is mostly concerned with elicitation and analysis of user requirements, deferring the activities of requirements specification and validation to be undertaken by the commissioner in a future work. In practice, the requirements development activities are interleaved and executed in incremental and iterative fashion [23]. Requirements should be progressively refined during each iteration from higher-level into more detailed and precise requirements [23].

Requirements management is defined by Hull et al. [24] as “the process that captures, traces and manages stakeholder needs and the changes that occur throughout a project’s lifecycle.” Requirements management may be done with specific tools that traces each requirement with statuses like: proposed, in progress, drafted, approved, implemented, verified, deferred, deleted or rejected [23]. This supports the iterative nature of most software development.

The requirements are subject to change also due to the changes in the operating environment [25]. In network testing, these changes could be the development of new network technologies or advancements in post processing techniques that changes the way the data is collected. It is also common that a diverse stakeholder community might produce different requirements that might be conflicting or contradictory and must be prioritized [25].

3.1 Requirements

There are multiple levels and types of requirements, with various definitions. Software requirements are a subset of system requirements that cover also other possible support elements such as hardware, firmware, information and services [4]. SWEBOK states that “At its most basic, a software requirement is a property that must be exhibited by something in order to solve some problem in the real world.” Another, more detailed definition of requirements is provided by Voas [26]:

> Requirements are specification of what should be implemented. They are descriptions of how the system should behave, or of a system property or attribute. They may be a constraint on the development process of the system.

The descriptions of how the system should behave are also generally referred to as functional requirements. Functional requirements describe what the system or a software should do, or they might explicitly state what it should not to do [25]. They can be written in different levels of detail, describing how the system or a software should behave in particular situations and react to particular inputs [25].

The descriptions of a system property or an attribute are generally referred to as nonfunctional requirements. They act as a constraint for the system or the software and are often concerned with quality metrics like performance, usability, maintainability, safety, reliability, security, or interoperability [4]. A well specified requirement is testable or measurable to allow validating the software or a system against it [24].

Terms “user requirements” or “stakeholder requirements” are also sometimes used in the field of requirements engineering with various definitions. By Wiegers and Beatty [23], “user requirements describes goals or tasks the user must be able to perform with the product that will provide value to someone.“ By this definition, user requirements are at higher level of the functional requirements that are derived from them.

3.2 Requirements Elicitation

Requirements elicitation is typically seen as the first phase of requirements engineering, where an understanding of the problem to solve is developed [27]. In iterative requirement and software development processes, it also includes understanding the existing requirements that might have been already implemented. According to Wiegers and Beatty [23], “elicitation is a collaborative and analytical process that includes activities to collect, discover, extract, and define requirements”.

Before starting the elicitation process, it’s beneficial to identify different user classes for the project, to be able to discover and capture unique needs that they have [23]. Understanding these needs and eliciting the requirements is challenging. Stakeholders may describe their needs in various ways and have difficulties to articulate what they want, or they are using jargon and expecting a deep knowledge in the domain [25]. In a dynamic economic and business environment, the requirements and their importance keep changing and new requirements appearing [25].

The most fundamental requirements elicitation techniques are interviews and observation or ethnography [25]. These techniques are used to understand the stakeholders’ needs and workflows. Requirements, especially for existing systems, may also be elicited by analyzing the system’s interfaces, user interface or existing documentation [23].

Individual or small-group interviews can be facilitated to ask stakeholders directly about their needs for the software system [23]. Interviews may be closed interviews with predefined set of questions or open interviews without predefined agenda, or mixture of both [25]. Some interviewing practices include, suggesting (alternative) ideas for a functionality, identifying use cases [23], discussing scenarios, asking why-questions, and recording the interviews [24]. The interviewer should be open minded and actively listen to the stakeholders without judging [23], [24], [25].

Ethnography helps to better understand how the social and organizational factors affect the operation of a system [25]. By observing an users perform their task using the software system, new requirements may be elicited that would have been difficult for the users to describe or articulate [23], [25]. Observation can be done silently or interactively, interrupting the user for questions mid-task [23]. Observation can be used with prototypes or understanding an existing product, but it might limit innovation that requires out-of-the-box thinking [25].

3.3 Requirements Analysis

Requirements elicitation is followed by requirements analysis. In this phase, the elicited requirements are categorized, high-level requirements are decomposed into appropriate level of detail and functional requirements are derived from the user requirements [23]. Quality attributes are identified for nonfunctional requirements [23]. Conflicts between the requirements are detected and resolved and the requirements are prioritized during the analysis phase [4].

Gaps in the requirements that are identified in the requirements analysis phase may be covered by iterating the requirement elicitation process [23]. Various visual requirements models may be developed to support the requirements analysis [4], [23]. Examples of these models are use case diagrams, data flow diagrams and data models, which could be documented using the Unified Modeling Language (UML) [4].

3.4 Applicability to This Study

The three phases of this study, as described in the methodology in chapter 2, are all implementations of requirements engineering processes, mostly requirements elicitation and analysis. They are part of an iterative requirement engineering process for the existing Nemo software.

The quantitative component discussed in chapter 5 is concerned with the data requirements of Nemo collection tools. While the format of the data is specified, it’s quantity in practice must be studied in practice. This information is helpful when specifying and validating the requirements for software components handling the data.

The primary focus of the study is on the user requirements related to data representation in Nemo collection tools. Requirements elicitation techniques of user interface analysis and document analysis are used to elicit current data representation capabilities and requirements of Nemo Outdoor (section 4.3) and Nemo Handy-A (section 4.4), which are then analyzed in Appendix 1.

Finally, the user requirements for data representation in Nemo collection tools are elicited using interviews, which is one of the most common techniques in both requirements engineering and qualitative research. The interviews also touch on requirements validation as the current capabilities of the software are discussed with the stakeholders. The requirements analysis is done in chapter 6.

4 Nemo Wireless Network Solutions

The development of GSM stared in 1980s and the first successful call was achieved by Nokia in March of 1991 [28]. The first public demonstration of the technology in July of 1991 still had to fallback to an analogue backup system [28].

Nokia Cellular Systems, developing technology for GSM networks, discovered that they don’t have a technology to measure live GSM networks [29]. They contacted another Finnish technology company, Elektrobit, to develop such a system [29]. The system was based on Nokia’s NMS/X tool (see Figure 2) and used Nokia 6050 car phone as a test terminal, the same model that was used in that first public demonstration [29].

Figure 2. Nokia NMS/X 3.0 [29].

In 1996 Nokia and Elektrobit founded a joint venture called Nemo Technologies to further develop network measurement and analysis solutions [29]. Initially the measurement products were based on technology licensed from Nokia. As network technologies evolved and importance and usage of mobile telecommunications increased, the measurement solutions expanded support for other vendors, too. The customer base also grew with mobile network operators and network equipment manufacturers around the world.

In the November 2006 Nemo Technologies was sold to UK-based public limited company Anite Group Plc. [30], that changed its name to Anite Plc. the following year. Anite Plc. was in turn acquired by US-based electronic measurement company, Keysight Technologies, Inc., in the August of 2015 [31]. Within Keysight, the network measurement business is called Nemo Wireless Network Solutions or Nemo Wireless Solutions [15], [32]. The products still carry the Nemo name.

This study focuses on the network measurement tools, while Nemo product line has also included products and solutions for analyzing the measurement, creating reports and managing the automated measurement fleet.

4.1 Product Evolution of Nemo Collection Tools

Collection tools can be divided into two categories: PC based and handheld tools. Scale of PC based measurement solution can vary from a laptop with a single measurement terminal to a specialized hardware installed into a rack in a laboratory or into a drive test vehicle. Handheld tools range from a single measurement terminal to a set of measurement devices that can be carried by a single person in a backpack type solution. A measurement solution typically consists of one or more mobile phones and could also include a network scanner. The measurement device can also be some other network device, like an IoT dongle. Some measurement solutions are designed to work autonomously without an operator physically interacting with the system [33].

The evolution of Nemo collection tools is shown in Figure 3. The PC based collection tools evolved from NMS/X to TOM to Nemo Outdoor. These products are described in more detail in chapter 4.1.1. Handheld collection tools were first introduced with NIB, which lead to the development of Nemo Handy product line. These products are described in more detail in chapter 4.1.2.

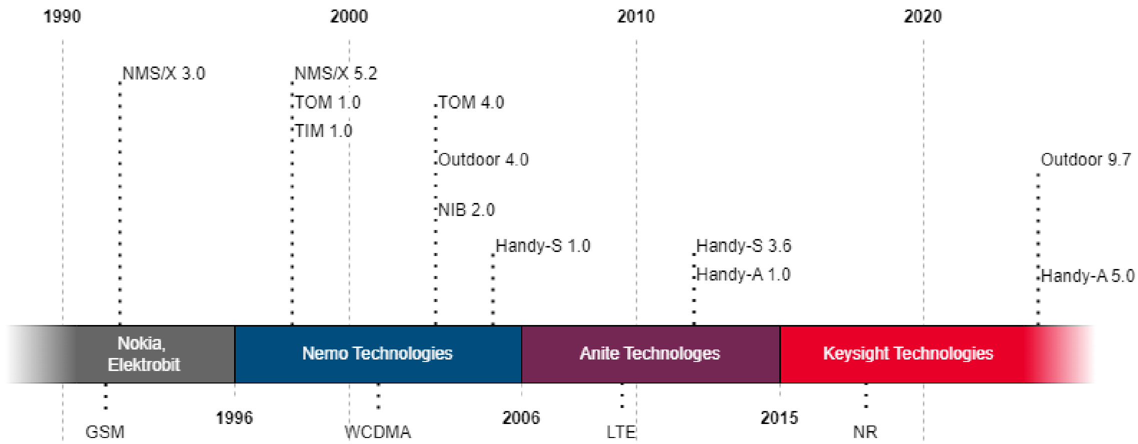


Figure 3. Evolution of Nemo collection tools [33], [34], [35], [36], [37], [38], [39].

Figure 3 also shows when each of the major network technologies were first commercially available and the company responsible for the Nemo products.

4.1.1 Evolution of PC Based Collection Tools

The first collection tool platform was called NMS/X [34], and it was inherited from Nokia. It was running on Microsoft Disk Operating System (MS DOS) and used to measure GSM, DCS or NMT network on a single device. In its final form, NMS/X 5.2 supported up to four measurement terminals that were connected to a laptop computer on a Multi Mobile Adapter Cable (MMAC). GPS receiver could be added to the setup to record location data for drive tests. NMS/X 5.2 supported six different Nokia mobile phone models.

NMS/X was replaced by products called TOM (Tool for Outdoor Measurement) and TIM (Tool for Indoor Measurement) in 1998. Both run on Microsoft Windows 95 or Microsoft Windows NT environments. This enabled a user interface that was comprised of multiple different types of data windows as illustrated in Figure 4. The UI configuration could be modified by the user and saved into a file. This allowed configuring different data views for each use case and sharing the configurations with other users.

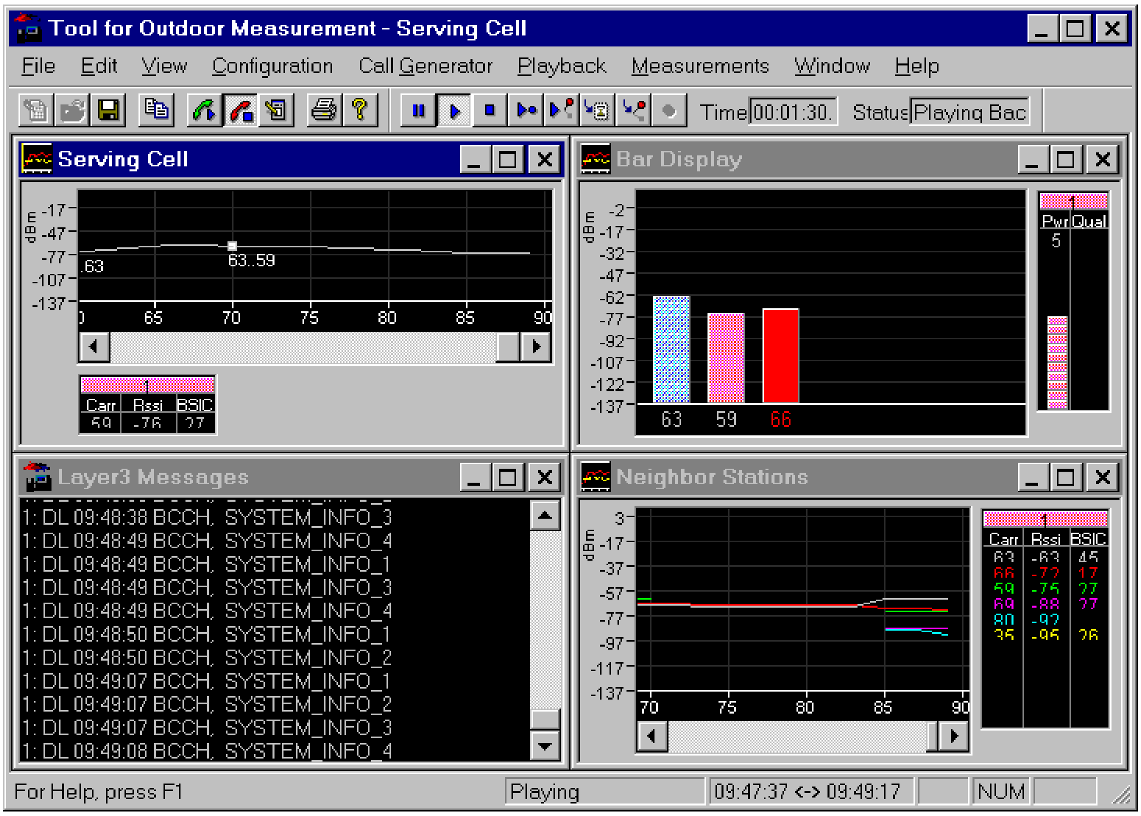


Figure 4. Tool for Outdoor Measurement (TOM) user interface. [35]

By 2003 TOM had reached its 4th major version and it was rebranded to Nemo Outdoor, keeping the version number 4. It supported GSM mobiles from Nokia and Sagem, GPRS mobiles from Motorola, Nokia and Sagem, single HSCSD mobile from Nokia as well as single WCDMA mobile from Nokia. Nemo Outdoor 4 supported also WCDMA scanners from DTI and Anritsu. [37]

Development of Nemo Outdoor has continued since with the evolution of mobile network technologies such as LTE and NR. The latest version as of writing this report, Nemo Outdoor 9.70, was released in February 2024.

Nemo Outdoor has been used as the underlying software for many derivative network measurement solutions that have used additional hardware such as backpacks, racks, or other enclosures for the measurement terminals. Nemo Outdoor has also been offered as a playbacking software for the measurements made with Nemo Handy-A, a handheld network measurement application, described in more detail in following chapters. Data representation capabilities of Nemo Outdoor is introduced in more detail in chapter 4.3.

4.1.2 Evolution of Handheld Collection Tools

Nemo Technologies' first handheld collection tool was called NIB (Nemo Intelligent Battery) [36]. The NIB system was based on compliant Nokia 6150 or 6190 GMS mobile, which battery was replaced with a specialized hardware, called the NIB battery unit. The test unit was connected to a Windows PC with a data cable for configuration with a utility software. During measurement, NIB was controlled with keypad commands on the mobile. NIB Solo 2.0 was released in 2002.

In 2005 Nemo Handy was released. The first version was running on Nokia mobile phones with Symbian operating system and was called Nemo Handy-S. The software logged diagnostic data from the device it was running on. The data was also represented in the user interface as illustrated in Figure 5.

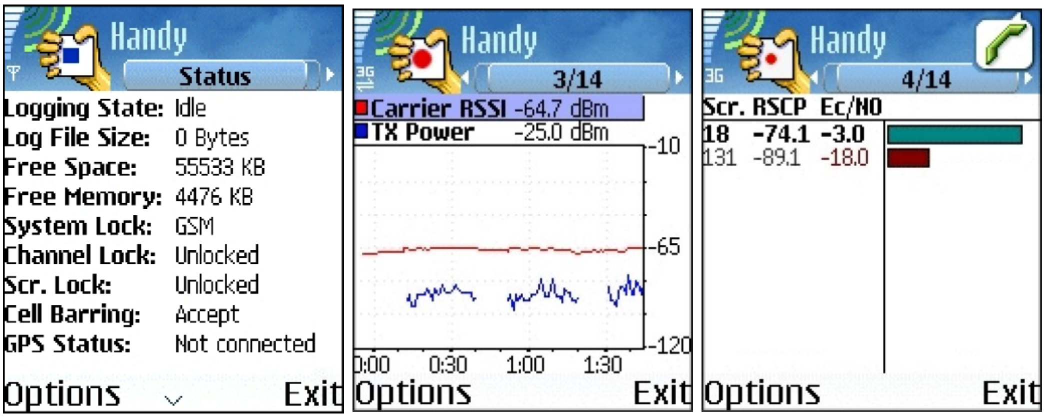


Figure 5. Nemo Handy-S user interface. [38]

Windows Mobile based Nemo Handy-W was launched in 2010 with support for HTC Touch Pro2 and HTC HD2 smartphones with GSM and UMTS networks [40]. This product had a short lifespan as the market shifted towards Android and iOS smartphones. In response to this market shift, Android based Nemo Handy-A was released in 2011. The latest version as of writing this report, Nemo Handy-A 5.00, was released in March 2024.

Multiple product variants have been based around Nemo Handy-A, most notably Nemo Walker Air, which allows multiple measurement terminals, also known as Agents, to be connected to a single Controller device for simultaneous measurements. Data representation capabilities of Nemo Handy-A is introduced in more detail in chapter 4.4.

4.2 Use Cases

The intended use cases for network measurement tools fall into two main categories: network and network device related measurements, as the collection tools such as Nemo Outdoor and Nemo Handy-A measure how a network device acts as a part of the network. Both categories can be divided into performance and benchmarking use cases. Many use cases require additional data processing and analysis, while some can be completed with the collection tool alone.

Typical customers for network collection tools are network and smartphone vendors, network operators and network benchmarking companies. Collection tools may be used in a lab or in a field. As an example, collection tools can be used in development and rollout of new network features. They can be also used in network optimization or benchmarking.

4.3 Data Representation in Nemo Outdoor

The UI in Nemo Outdoor is built around the concept of a **workspace**. A workspace is a UI configuration that can be saved to and loaded from a file.

Each workspace is composed of one or more **view groups** that are presented as tabs in the bottom of the UI. Each view group may contain multiple **data windows**, which can be arranged and resized by using a drag and drop functionality as illustrated in Figure 6.

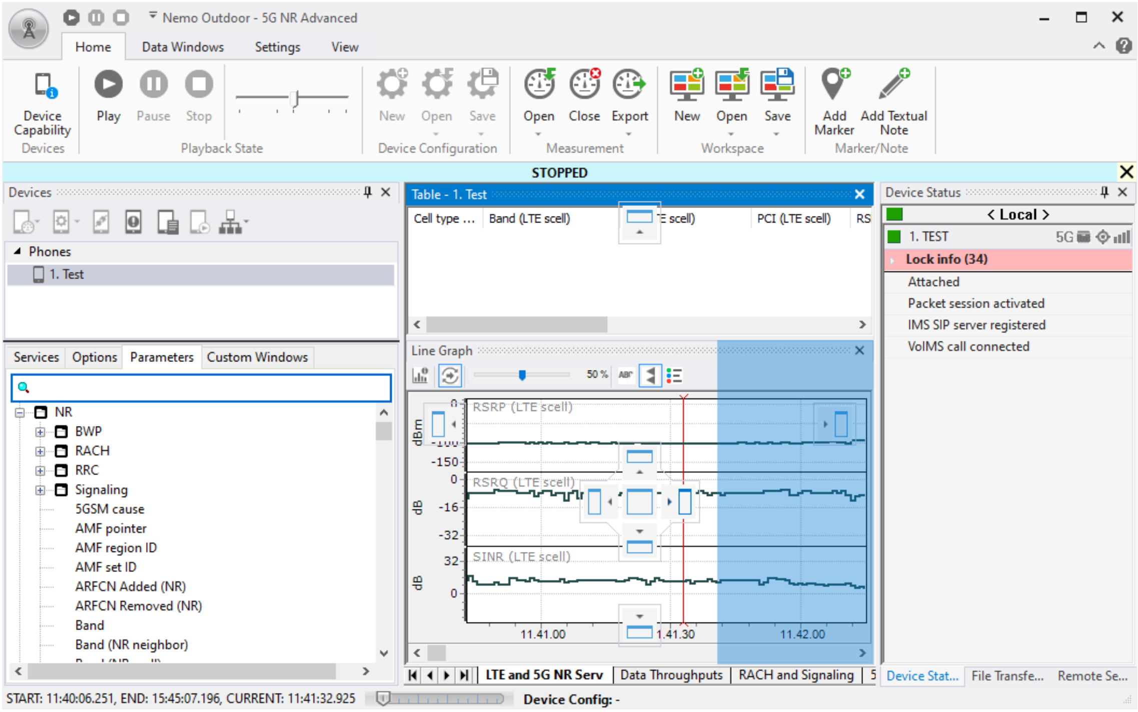


Figure 6. Nemo Outdoor user interface.

Data windows in Nemo Outdoor are categorized into four categories: graph, grid, indoor and map. The user can also save a configured data window as a custom window, which can be then opened into a view group on a workspace. Empty data windows, or data windows with preconfigured parameters can be added from a context menu that is opened by right clicking an empty view group. Similarly, these data windows are available from the ribbon bar on Nemo Outdoor.

Nemo Outdoor also has a parameter view that lists all available parameters in a tree structure with a search functionality (see bottom left in Figure 6). Each parameter has a set of compatible data windows it can be added to. Multiple parameters can be selected simultaneously from the parameter tree and opened to a new data window by right clicking and selecting one of the supported data window types from the context menu. Selected parameters can also be added to an existing data window using the drag and drop functionality, if the selected parameters are compatible with the target data window.

Data windows on Nemo Outdoor are synchronized to represent the values around the same timestamp. When a value is clicked on any data window, all other data windows are updated to represent values around the timestamp of the selected value. If the selected timestamp is out of view, graph and grid type data windows are also automatically scrolled to the active position.

4.3.1 Graphs

Nemo Outdoor supports seven types of graph views: line graph, vertical bar graph, horizontal bar graph, scatter graph, spectrum graph, color grid graph, and gauge graph.

Each graph data window is composed of the graph area and additional supporting view elements that can be enabled or disabled from the context menu. These are: toolbar, scrollbar, axis, legend, side panel, layers, values, and extended value lists. The side panel may contain tabs for layers, values and extended value lists. Additionally, these views can be rearranged within the data window using the drag and drop functionality. Graph views can be saved as image from the context menu.

Parameters for the graph are configured in layers view. It also supports grouping layers. Layer groups are drawn to separate graphs that are stacked in the graph area, sharing typically the x-axis. This allows configuring parameters with different y-scale to be compared within a graph. Parameters that are in the same group share the y-axis on the graph. If the scales for the parameters differ, the scale of the active (selected) parameter is drawn.

Values view is a grid with columns for the parameter name and its value. Most graph types also support a legend that represents the same information overlayed on the graph. Alternatively, extended value lists can be enabled by parameter to show its value in a separate grid view with related parameter values.

Line graph supports multiple options for formatting the lines, such as changing the width, drawing as area, using color set or a fixed color, and averaging. Line can also be drawn as stairs, where the line is drawn horizontally between each value. Figure 7 illustrates line graph with two layer groups.

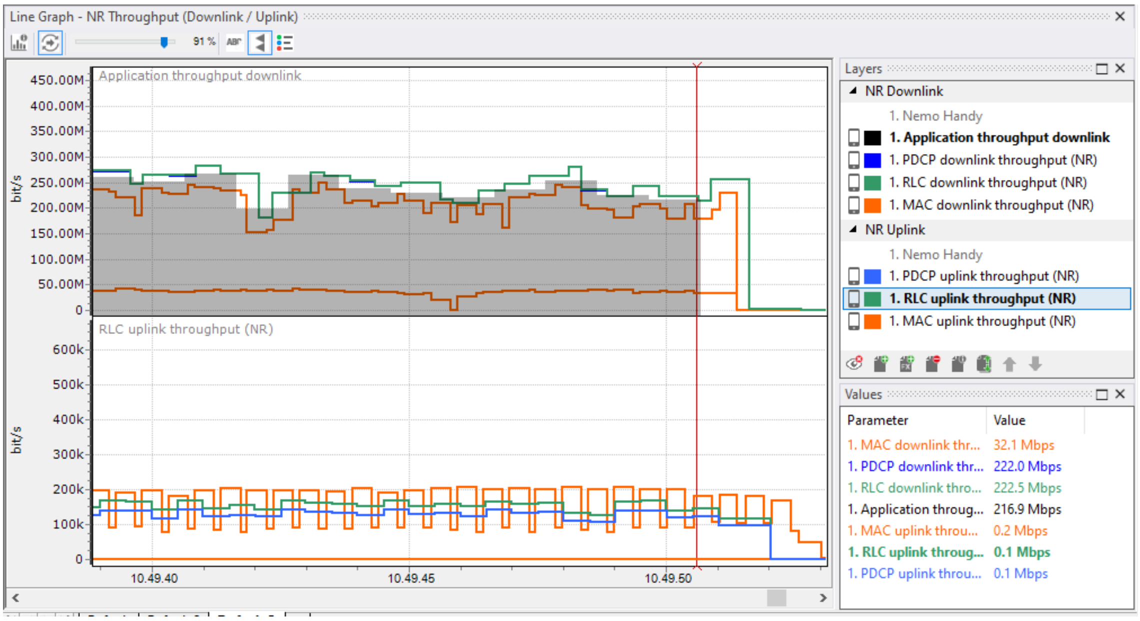


Figure 7. Line Graph in Nemo Outdoor.

Vertical bar graph and horizontal bar graph are equal in functionality, the difference being the orientation of the bars (see Figure 8). When using layer groups, horizontal bar graphs are stacked horizontally, sharing a common y-axis. Colors for the bars can be configured similarly to the lines in a line graph. Bar graphs supports sorting values in ascending or descending order. Optionally a label with the numeric value can be added for each bar.

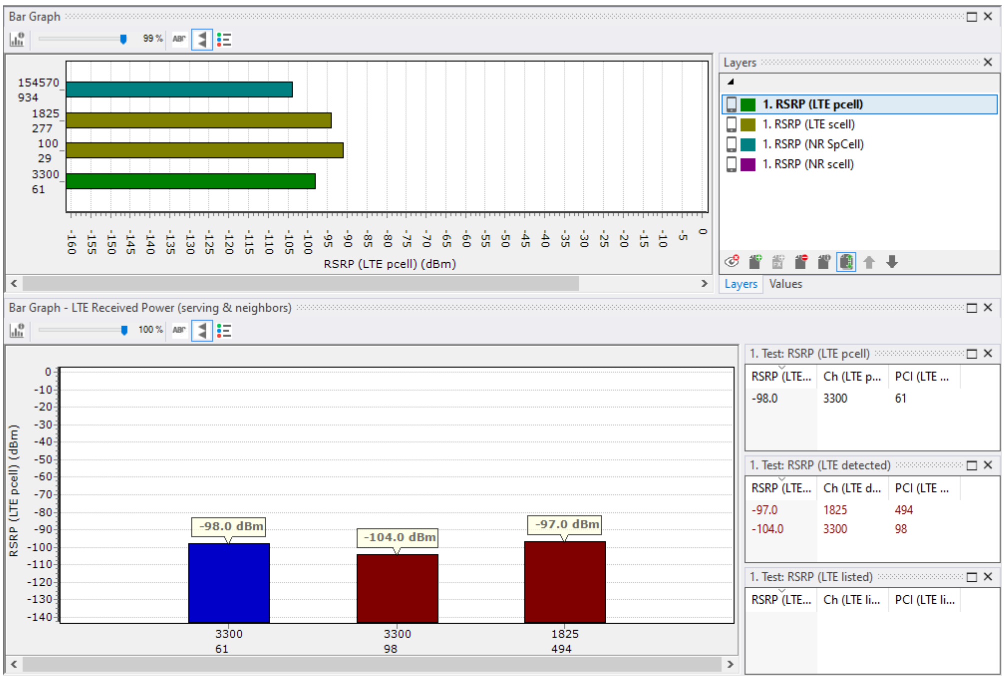


Figure 8. Horizontal Bar Graph and Vertical Bar Graph in Nemo Outdoor.

Scatter graph can be used to illustrate correlation between two parameters (see Figure 9). It cumulates data from the beginning of the measurement. Size, shape, transparency and color of the points can be configured.

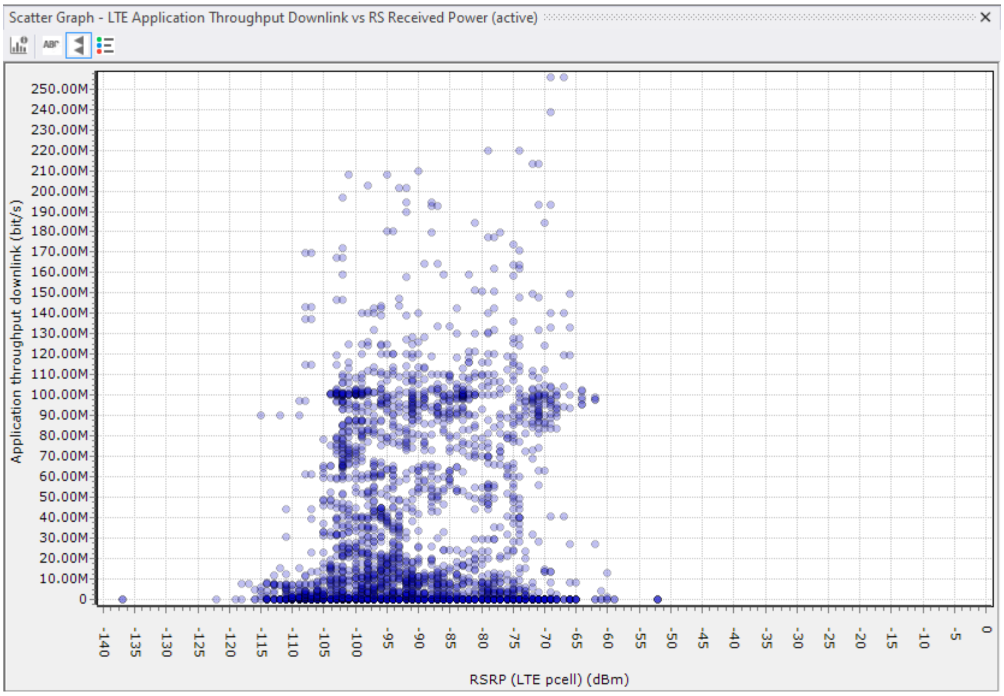


Figure 9. Scatter Graph in Nemo Outdoor.

Spectrum graph in Nemo Outdoor is illustrated in Figure 10. Spectrum graphs line can be configured similarly to a line in the line graph. Additionally, spectrum graph supports a peak line, which can be reset at any given time. Markers can be added by clicking the graph area, or from the edit marker window.

Figure 10. Spectrum Graph in Nemo Outdoor.

Color grid graph is mostly used as a spectrogram as the x-axis represents time and the parameters value is represented with color (see Figure 11). Custom color set can be configured for the parameter in the color grid graph. This graph type packs the most data in a single graph.

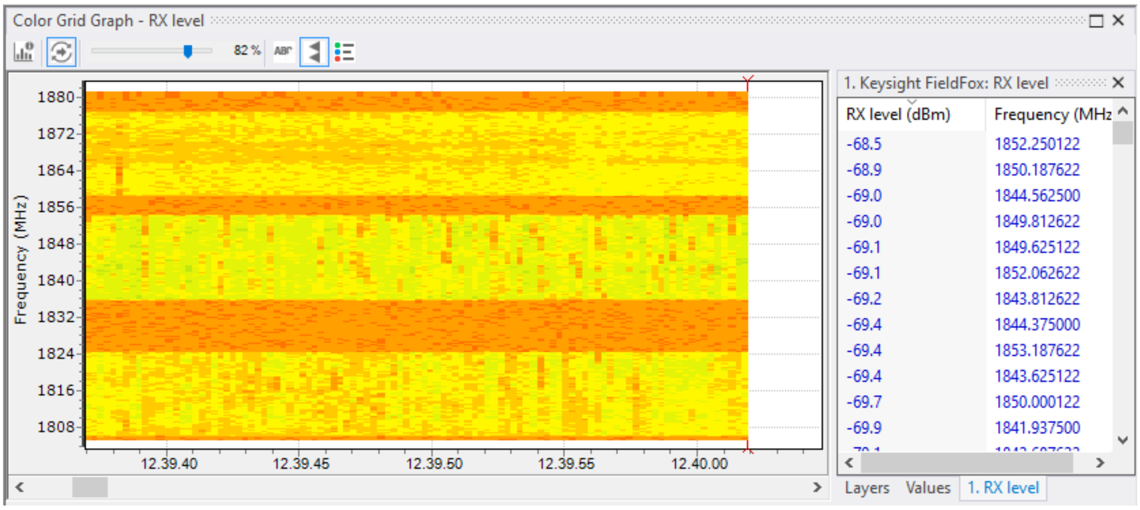


Figure 11. Color Grid Graph in Nemo Outdoor.

Gauge graph represents the values with an analog gauge element (see Figure 12). The numerical value and unit are represented under the gauge. This is the only graph view that does not have axis and therefore the layer groups don't have any effect. Each parameter is drawn in its own gauge in the graph area. The gauges are stacked either horizontally or vertically, depending on the available space.

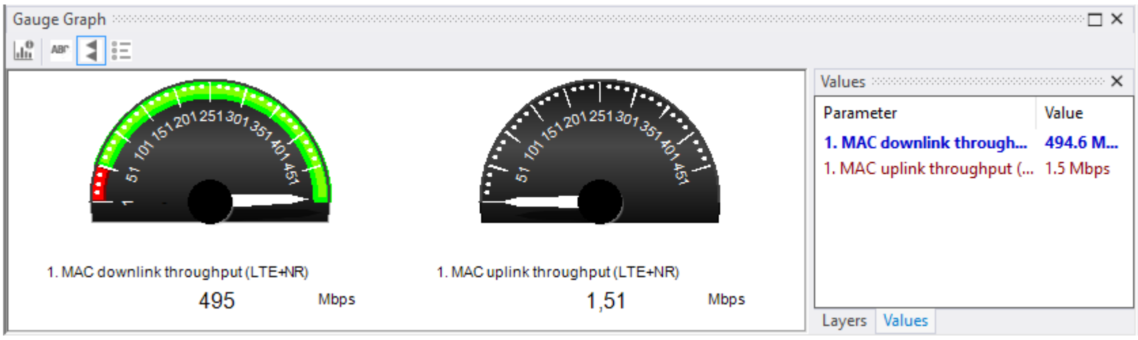


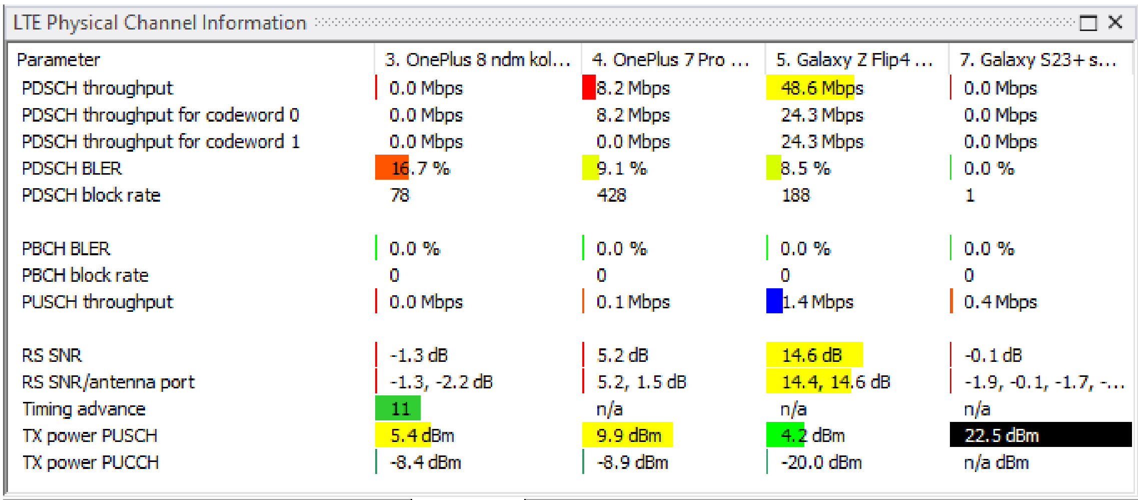
Figure 12. Gauge Graph in Nemo Outdoor.

4.3.2 Grids

Nemo Outdoor supports seven types of grid views that present data in a tabular form: events grid, parameters grid, table grid, packet grid, trace grid, sampling grid, and notifications grid.

All grid data windows support font size selection between four sizes: small, default, large, and extra large. Rows on a grid can be selected with a mouse or a keyboard. Selected rows can be copied or exported in various formats. Detailed copy and export types are provided in Appendix 1. Some grid types also support filtering the data. Filtering rules can be configured in a dialog from a context menu, or in some cases quick filtering options are suggested in the context menu, based on the selected row.

Colors for the text and cell background can be configured for all grid types using color sets except for notification rows, which have always black text on a pale-yellow background. Some color sets are provided by default, and they can be also configured by the user. For numerical values, cell background is colored partially starting from the left edge, based on how the value intersects the range of the color set. This makes the column effectively a simple bar graph as illustrated in Figure 13.


LTE Physical Channel Information

Parameter	3. OnePlus 8 ndm kol...	4. OnePlus 7 Pro ...	5. Galaxy Z Flip4 ...	7. Galaxy S23+ s...
PDSCH throughput	0.0 Mbps	8.2 Mbps	48.6 Mbps	0.0 Mbps
PDSCH throughput for codeword 0	0.0 Mbps	8.2 Mbps	24.3 Mbps	0.0 Mbps
PDSCH throughput for codeword 1	0.0 Mbps	0.0 Mbps	24.3 Mbps	0.0 Mbps
PDSCH BLER	16.7 %	9.1 %	8.5 %	0.0 %
PDSCH block rate	78	428	188	1
PBCH BLER	0.0 %	0.0 %	0.0 %	0.0 %
PBCH block rate	0	0	0	0
PUSCH throughput	0.0 Mbps	0.1 Mbps	1.4 Mbps	0.4 Mbps
RS SNR	-1.3 dB	5.2 dB	14.6 dB	-0.1 dB
RS SNR/antenna port	-1.3, -2.2 dB	5.2, 1.5 dB	14.4, 14.6 dB	-1.9, -0.1, -1.7, -...
Timing advance	11	n/a	n/a	n/a
TX power PUSCH	5.4 dBm	9.9 dBm	4.2 dBm	22.5 dBm
TX power PUCCH	-8.4 dBm	-8.9 dBm	-20.0 dBm	n/a dBm

Figure 13. Parameters Grid in Nemo Outdoor.

Parameters grid (see Figure 13) is the only grid type that supports data from multiple devices to be represented in a single table, with each device allocated to a column. Multiple parameters can be added to parameters grid, each assigned to a new row. Parameters grid shows only the current values of the parameters.

Events grid represents measurement data from a single device (see Figure 14). Multiple parameters can be added to events grid as columns. Each measurement event that covers any of the parameters on the event grid are added as a new row. Rows are sorted in the ascending chronological order. Sampling grid is a variation of events grid, where each row is added with the specified time interval instead of the recorded measurement event.

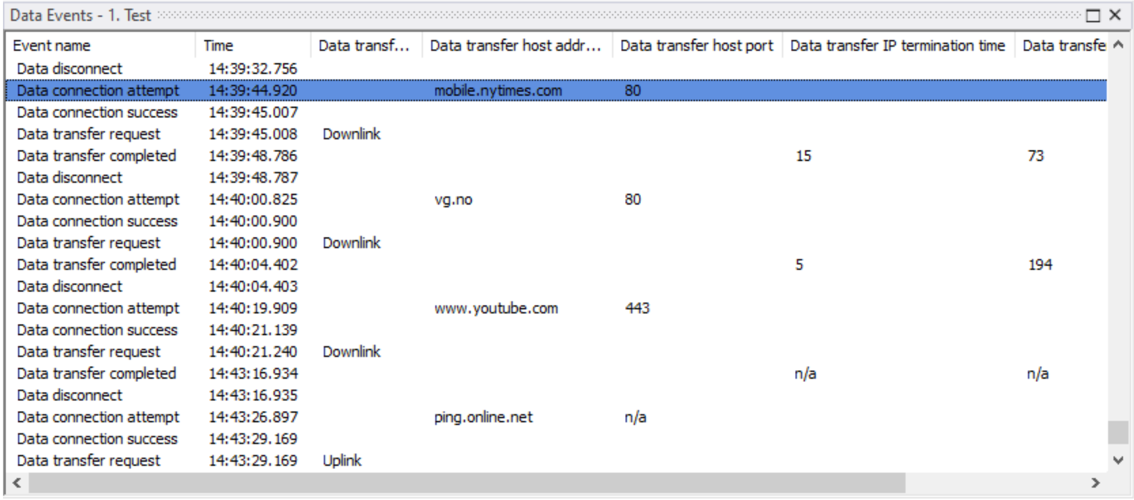


Figure 14. Events Grid in Nemo Outdoor.

Packet grid (see Figure 15) represents packet capture data. Each recorded frame is represented in a row in the grid. Packet capture frames can be automatically decoded in a sub-window on the packet grid data window, or the user may select a specific frame to be decoded.

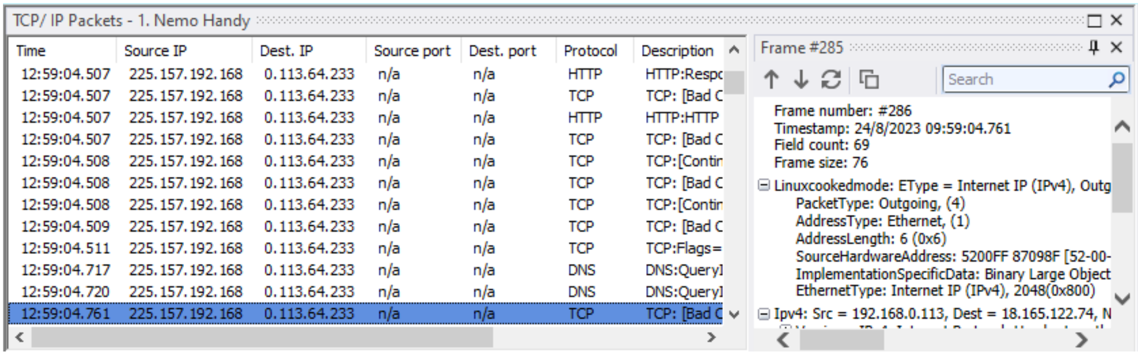


Figure 15. Packet Grid with decoded message in Nemo Outdoor.

Trace grid represents chipset specific raw trace data (see Figure 16). Data is presented in hexadecimal string format and can be decoded into a sub-window similarly to the packet capture frames in packet grid.

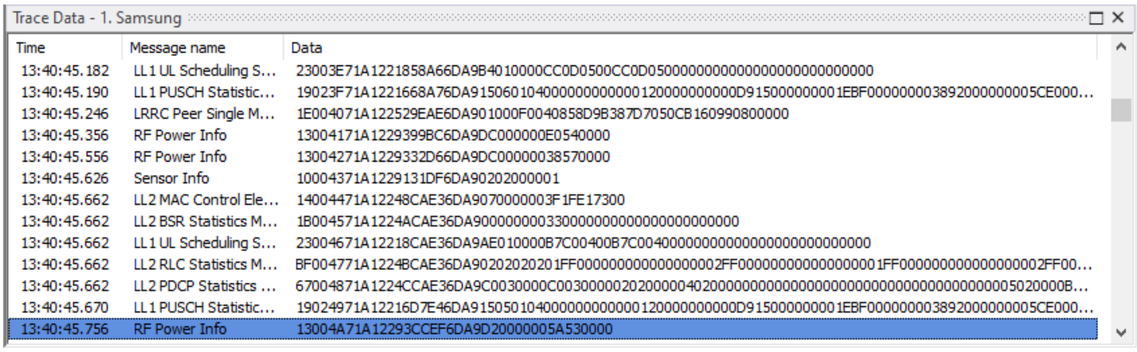


Figure 16. Trace Grid in Nemo Outdoor.

Notification grid represents all enabled notifications for a single measurement device. Notification grid has columns for notification name and timestamp. Optionally, notifications can be enabled for an event grid, where the notification rows are placed between the event rows.

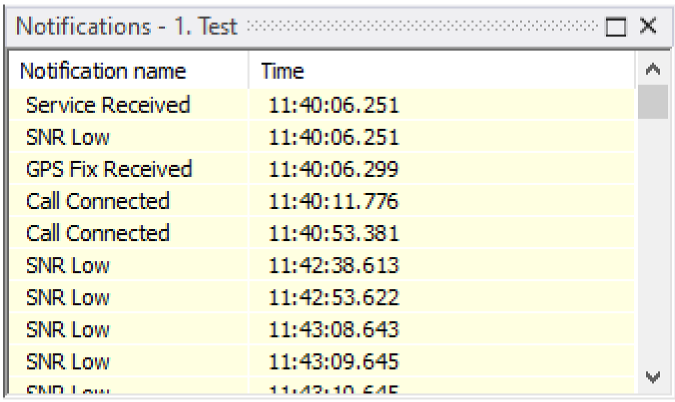


Figure 17. Notifications Grid in Nemo Outdoor.

Table grid represents the current values of related parameters for a single device (see Figure 18). Parameters allocated to columns can be selected from the properties dialog. Table grid is well suited to representing parameters, which contain list values, which are represented as comma separated values in parameters grid or events grid columns. Table grid allocates a new row for each entry in the list, making each column containing only a single value, therefore making it aligned with the concept of tidy data [41].

Table - 1. Test

Cell type (NR neighbor)	Band (NR neighbor)	NR-ARFCN (NR neighbor)	PCI (NR neighbor)	RSRP (NR neighbor)	RSRQ (NR neighbor)	SINR (NR neighbor)	RS
Listed or detected	NR n78 TDD	641280	45	-110.9	-14.0	0.8	[illegible]
Listed or detected	NR n78 TDD	641280	97	-113.1	-14.5	-0.4	[illegible]
Listed or detected	NR n78 TDD	641280	58	-117.0	-18.8	-6.9	[illegible]
Listed or detected	NR n78 TDD	641280	215	-117.2	-18.5	-6.6	[illegible]
Listed or detected	NR n78 TDD	641280	138	-117.4	-19.4	-7.5	[illegible]
Listed or detected	NR n78 TDD	641280	128	-118.7	-18.6	-6.6	[illegible]

Figure 18. Table Grid in Nemo Outdoor.

4.3.3 Indoor

Indoor map supports representing measurement results on top of MapInfo, iBwave and Google Earth maps. Indoor measurement routes are typically recorded with markers during the measurement. Multiple routes can be added on top of the indoor map and arranged with an offset. Each route can be colored for a single parameter of a measurement device. For example, different routes may have the same parameter for different devices (see Figure 19), or different parameters for the same device.

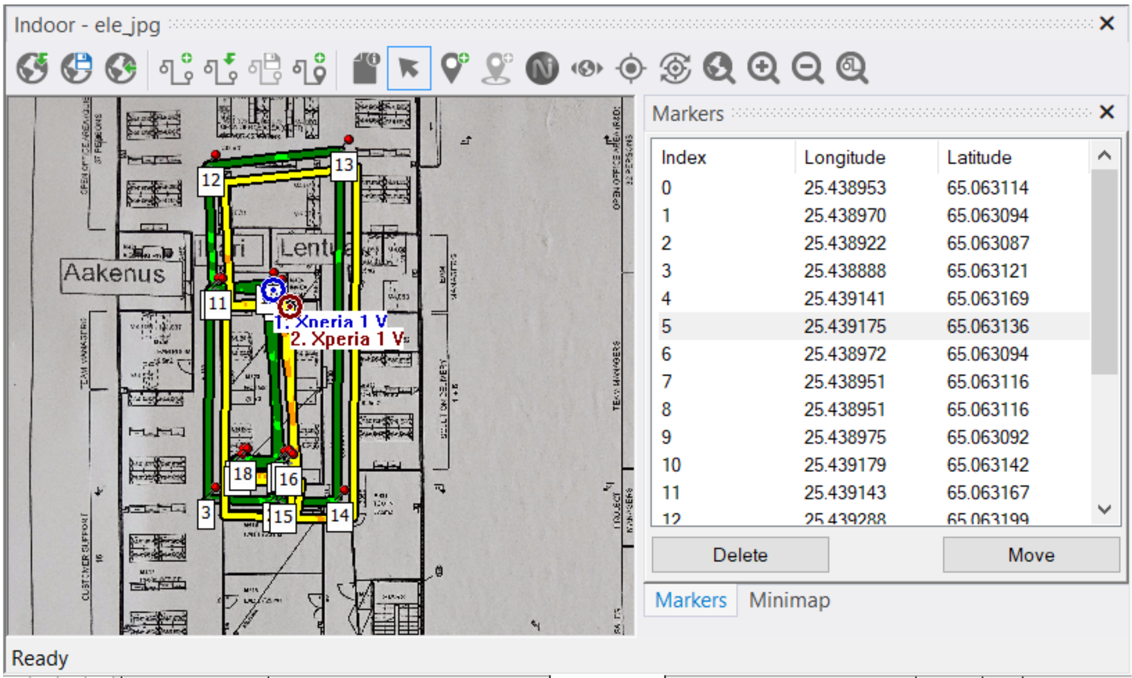


Figure 19. Indoor map in Nemo Outdoor.

4.3.4 Map

Nemo Outdoor supports representing drive test results on Google Map, Open Street Map (see Figure 20) and MapXtreme. The route configuration is like with the indoor map, but the route data is typically a GPS track. MapXtreme map in Nemo Outdoor also supports adding a floorplan layer for georegistered indoor maps. Map data windows also support overlaying base station information over the map.

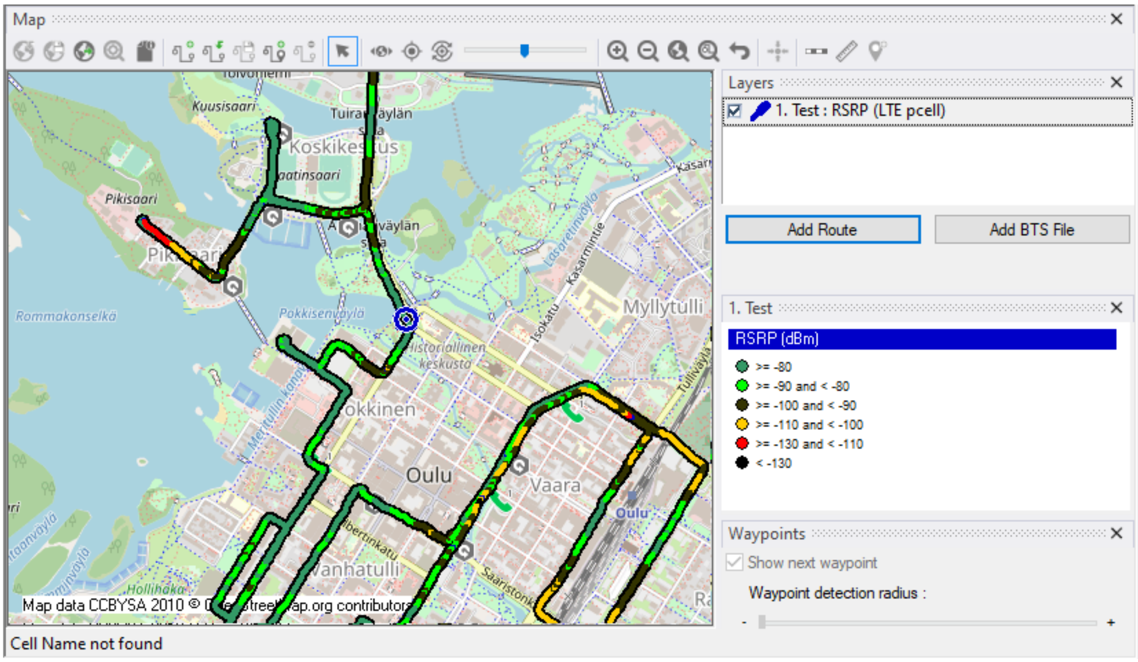


Figure 20. Open Street Map in Nemo Outdoor.

4.3.5 Notifications

Notifications in Nemo Outdoor are represented in notifications grid and event grid. Optionally, notifications can be represented with audio or with an icon on graphs and maps. The user can select any audio file in WAV format for the audio notification and any bitmap in PNG format for the notification icon. Some audio and bitmap files for notifications are bunded with Nemo Outdoor and used with the default notifications.

Notifications can be customized in Nemo Outdoor using the built-in Notification Manager and its Notification Criteria dialog. Notification criteria consists of one or more criteria, linked by AND or OR operators. Each criterion is comprised of parameter, operator and value. The use of Notification Criteria dialog is demonstrated in Figure 21.

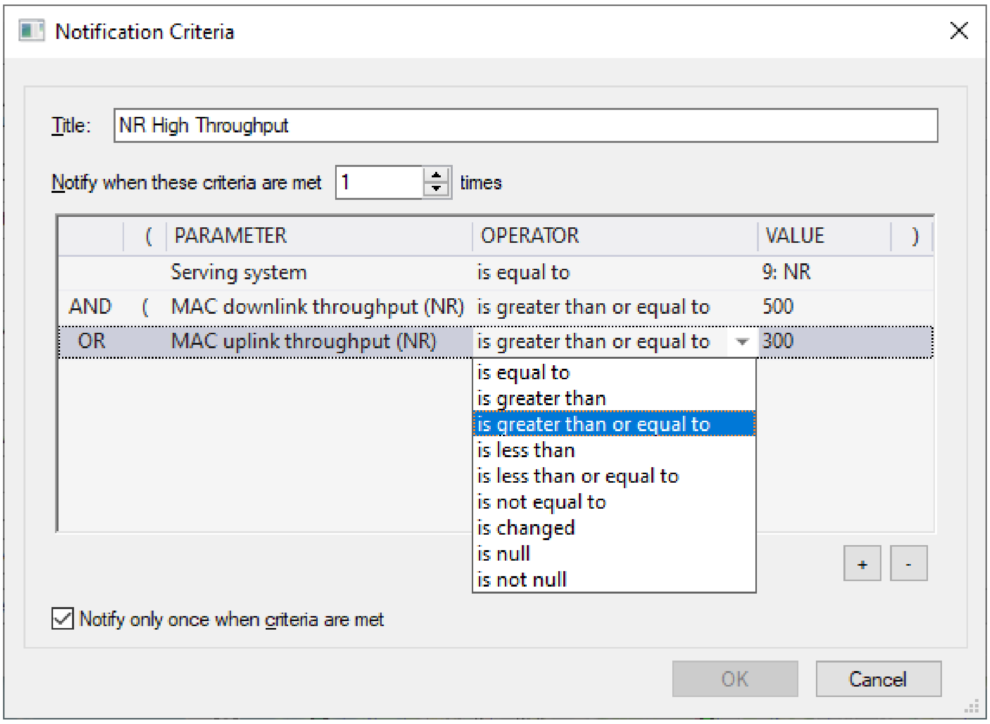


Figure 21. Notification Criteria dialog in Nemo Outdoor.

4.4 Data Representation in Nemo Handy-A

The UI in Nemo Handy-A is composed of multiple pages, also referred to as displays, that can be navigated by swiping horizontally, or selected from a dropdown list on the app bar. In this document, the term ‘display’ is used for a view that fills most of the screen and can be navigated as described earlier. In this document, the term ‘page’ is used for a view that is part of the display. The same design is used in Nemo Handy-A’s multi device variant, Nemo Walker Air.

A display may contain only a single page, or it may be split to contain up to five pages on a phone screen. On a tablet screen, six or eight pages are allowed, depending on the ‘Small Splits’ option. By default, the split area is divided equally for even number of pages. With the ‘Small Splits’ option, more space is allocated for the first page and the remaining space is divided equally for the remaining pages. Each split on a display may also contain multiple pages, which are navigated by swiping vertically.

Many displays in the default configuration are split into two pages, which are arranged in a column in the portrait orientation. It is also possible to rotate the device to landscape orientation, which causes the two pages to be arranged in a row. Most of the screenshots in this chapter use the landscape orientation for better alignment with the report’s format, while in practice Nemo Handy-A is typically used in the portrait orientation on a smartphone.

Each page is configured with an information about the mobile network system, on which the page shall be visible. The list of displays is updated dynamically to show only the displays that contain pages that match the active mobile network system of the measurement terminal.

Nemo Handy-A is delivered with some hardcoded displays and pages. It also has a default display configuration file that contain most of the displays available on the UI. Changes to the display configuration are saved to an XML file that may be shared between Nemo Handy-A users. The display configuration file contains also information about map parameters, notifications and ‘Serving Info’ parameters. These topics are explained in detail in following chapters.

4.4.1 Data Views

The pages on a Nemo Handy-A can be configured to display measured parameters in various formats. The basic page layout is a simple two-column grid with parameter labels on the left column and values on the right column (see Figure 22). The value also contains the unit, when available. The parameter can be configured so that its value is formatted with bold or cursive style. There’s an option to configure parameters to display their values in a simple bar graph element on a third column. The color of the bar is determined by a threshold value with good/bad colors or from a color set.

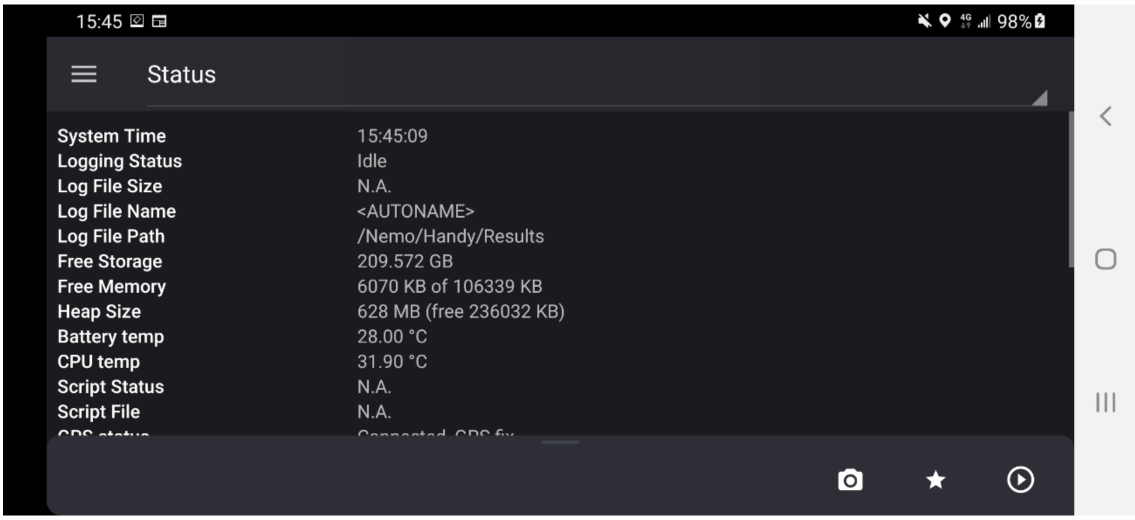


Figure 22. Status page in Nemo Handy-A.

List type parameters can also be displayed in a data grid, which is typically used for cell measurements. When representing cell measurements, the data grid typically contains parameters to identify the cell such as band, channel and PCI for LTE. Additionally, there might be signal quality parameters such as RSRP, RSRQ or SNR. The table may contain a column that is used as a horizontal bar graph as illustrated in Figure 23.

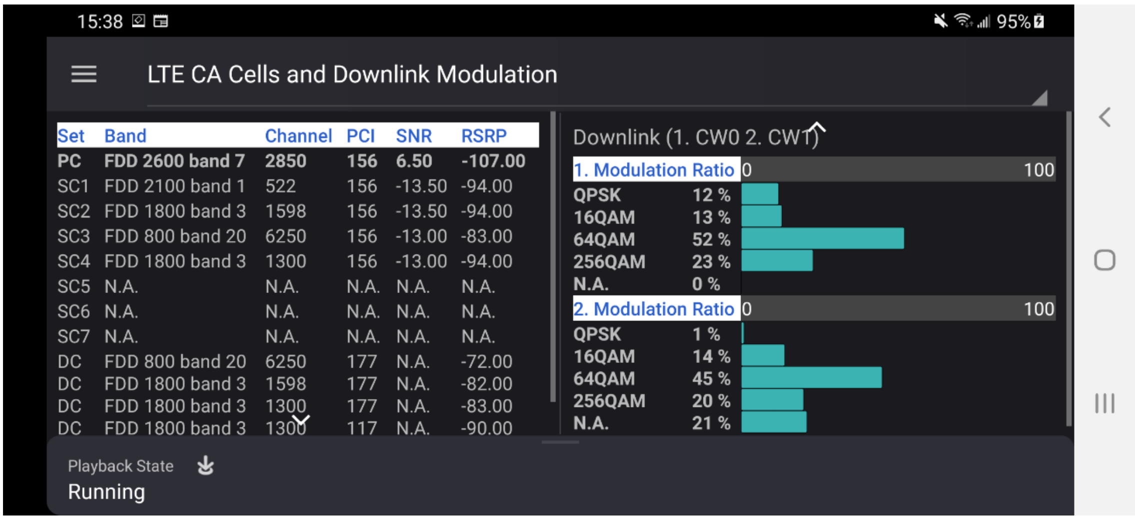


Figure 23. Data grids in Nemo Handy-A.

The most used data visualization technique in Nemo Handy-A is a line graph. Line graph can be configured in two ways into a page, as illustrated in Figure 24. When used with list type parameters, the graph component is on the right side of the data grid. When used with individual parameters, the graph component is below the parameters, presented in a label-value grid. On the latter line graph type, each parameter is scaled individually, and the graph component can be configured with a scale on both sides. Line graph with a list type parameter scales all the elements of the list with a single scale that is presented on the right side of the graph. Nemo Walker Air Controller also supports a special type of line graph that combines values from all the connected Agent terminals.

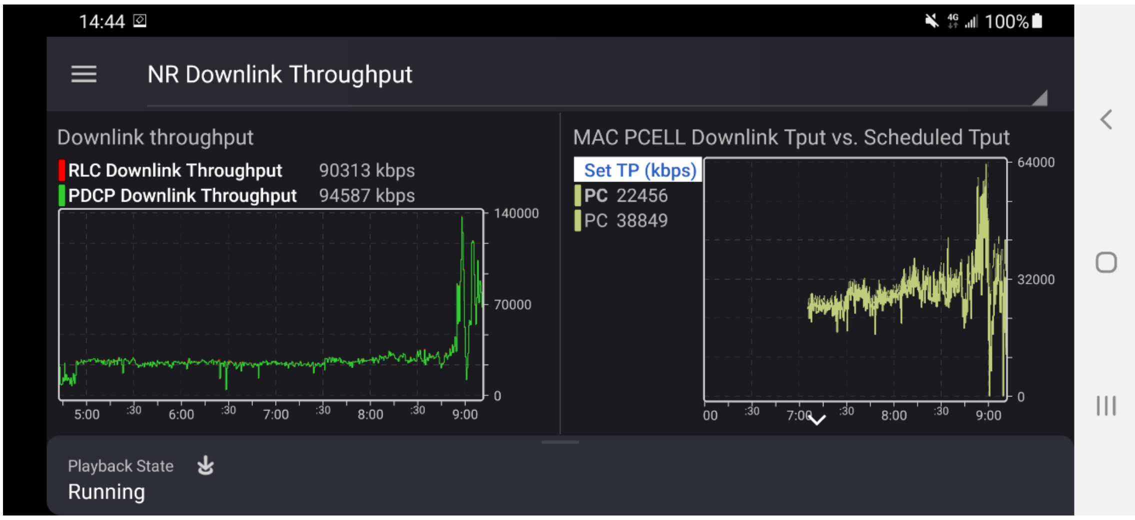


Figure 24. Line graphs in Nemo-Handy-A.

Another visualization component is an analog gauge. It is typically used to indicate the GPS speed of the measurement unit, or a throughput of a data transfer (see Figure 25).

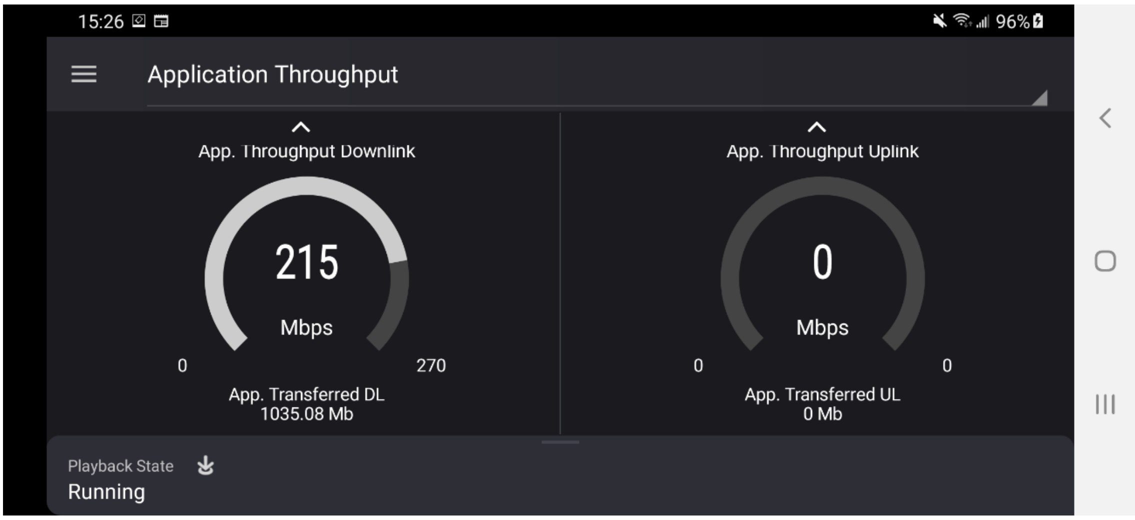


Figure 25. Gauge views in Nemo Handy-A.

Bar graph with vertical bars is also supported by Nemo Handy. Its main use case is to represent the signal levels of a network scanner measurement. The bar graph is also used for histograms for statistics as illustrated in Figure 26.

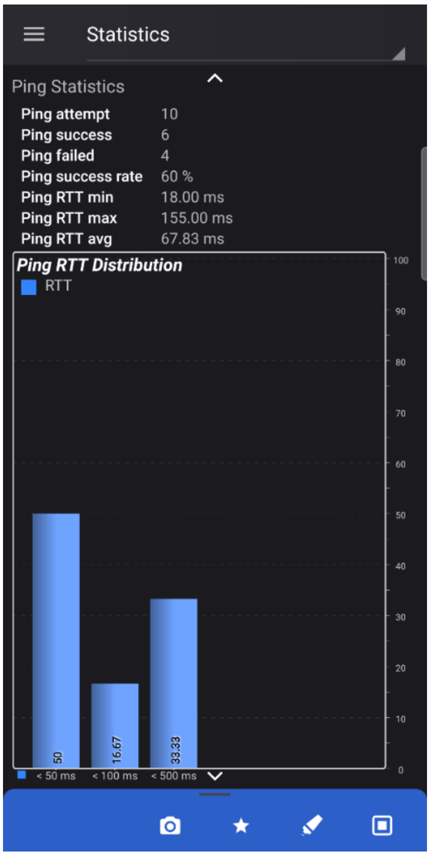


Figure 26. Ping Statistics in Nemo Handy-A.

WiFi Spectrum scan results can also be represented in a special spectrum scan graph component (see Figure 27).

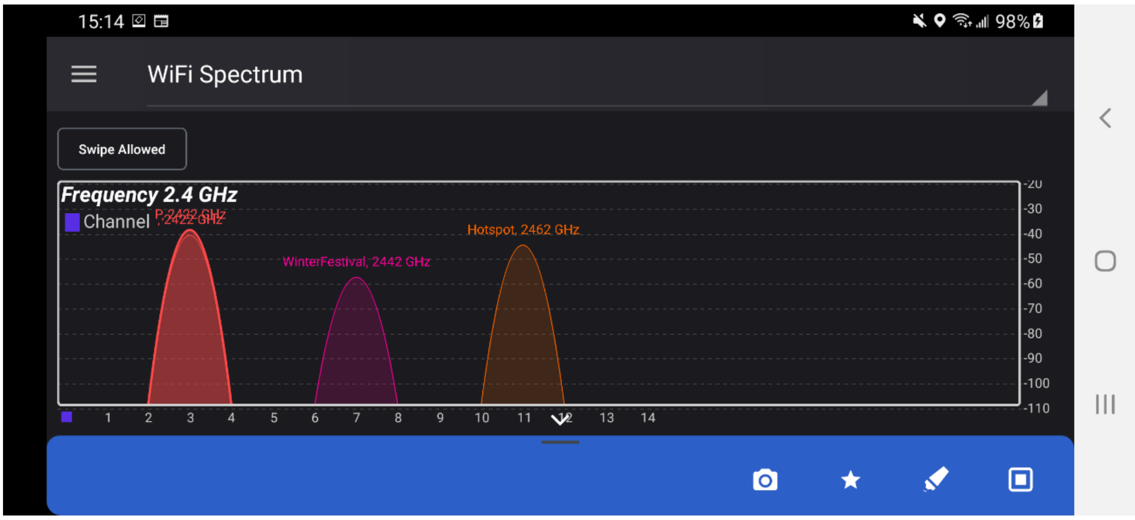


Figure 27. WiFi Spectrum graph in Nemo Handy-A.

4.4.2 Special Pages

Nemo Handy-A supports also multiple special page types that can't be configured in the display configuration file. These special pages can be classified into three categories: data, configuration and transaction.

Transaction pages are required for some test types. For example, the HTTP Browsing transaction uses the special Browser page to load the web page contents. The page is not used to represent the KPIs that are collected in the background such as connection attempts, successes and failures. Therefore, these pages are not relevant for this study.

Configuration pages are used to view or change the measurement configuration. Example of configuration pages is the Script page, which represents the state of the loaded or running script. Another example is Nemo Cloud page, on which the user can log in to Nemo Cloud to view the active project details.

Special pages for data representation include ‘Map’, ‘Notification History’, ‘Signaling’, ‘Transaction Log’ and ‘mScore Score Test Results’ pages. Except for Map page, these pages are composed of a scrollable list with custom layout.

Map page has a full-size Goole Map or Open Street Map view. Map location, rotation and zoom level can be controlled with one- and two-finger gestures. Current central coordinates, rotation and zoom level are overlayed in the top left corner of the page. Tapping the compass icon will reset the rotation to 0 degrees. When an indoor map floorplan is selected, it is overlayed on the world map. Depending on the measurement, either GPS route, route plan or a marker route is overlayed on the map and colored according to the active map parameter, which is presented in the legend box on the bottom left corner of the page. Map page also supports overlaying geographic data from KML files and base station information from BTS files.

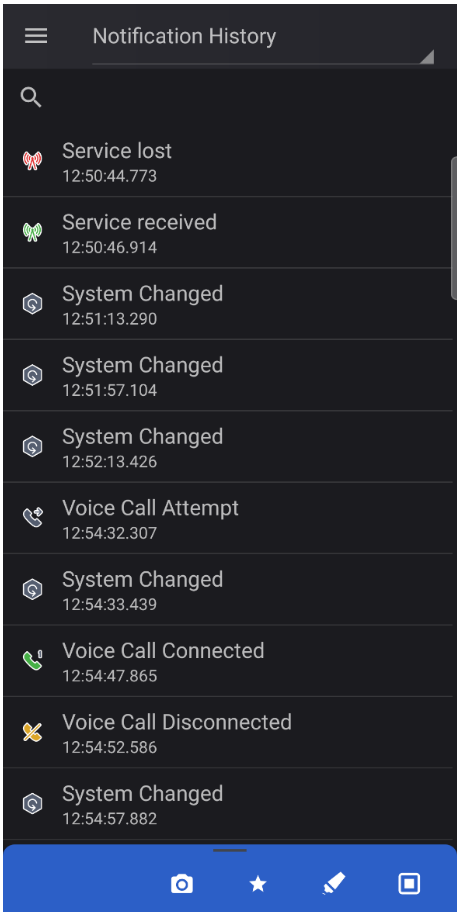


Figure 28. Notification History page in Nemo Handy-A.

Notification History page lists the history of triggered notifications (see Figure 28). Each item in the list is displayed with icon, title and timestamp. The list can be filtered using both inclusive and exclusive search terms.

Similarly to Notification History page, Signaling page lists the history of signaling messages with filter functionality (see Figure 29). The layout includes an icon for the direction (uplink or downlink), message header, timestamp and system. When a list item is tapped, a dialog is opened with the signaling message is represented in decrypted form and as hex data.

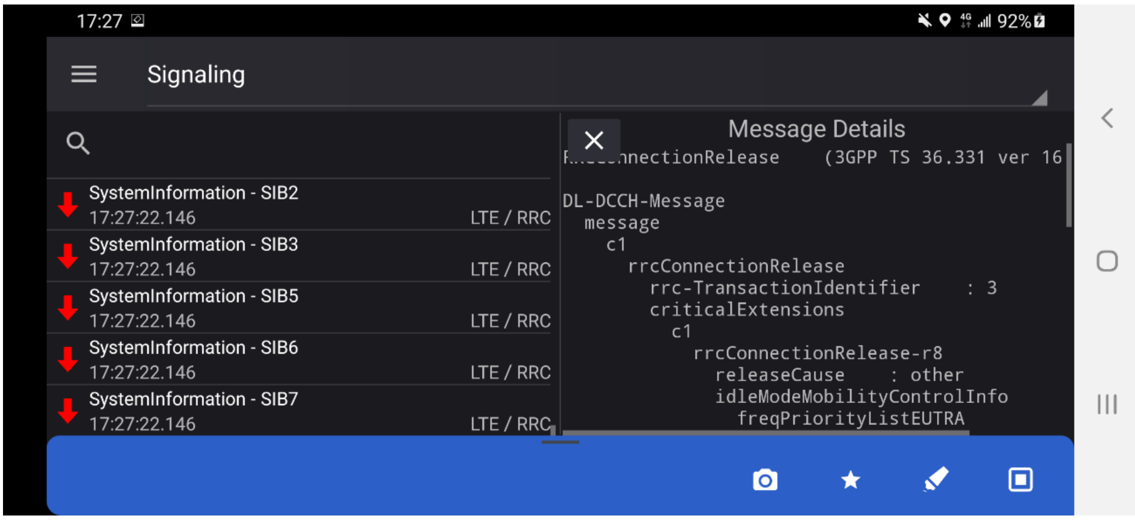


Figure 29. Signaling page in Nemo Handy-A.

Transaction Log page lists the transactions from the measurement (see Figure 30). Successful transactions are highlighted with green title bar and failed transactions with red title bar. The list can be cleared with the specific button on the top left corner of the page.

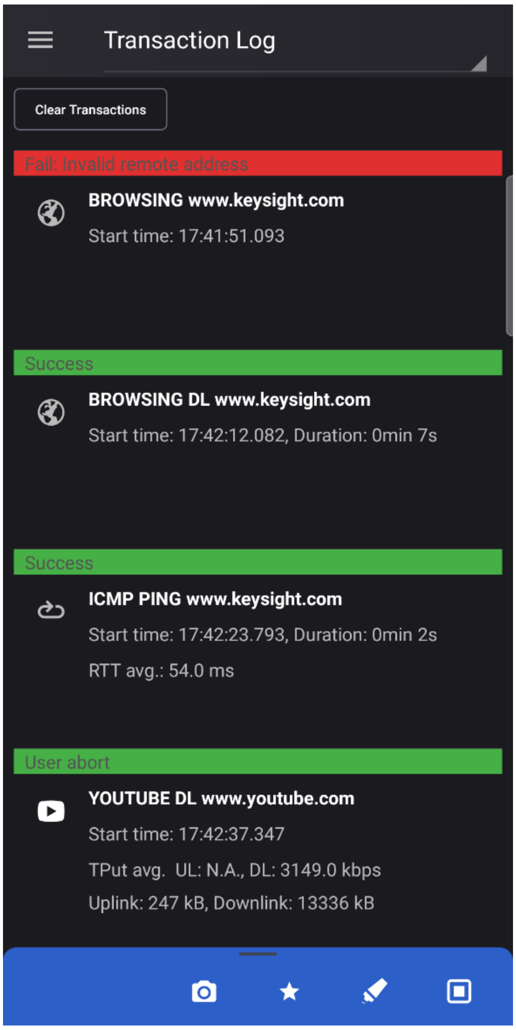


Figure 30. Transaction Log page in Nemo Handy-A.

Finally, mScore Score Test Results page lists detailed information about the performed mScore transactions (see Figure 31). Each item has a star rating up to five stars and numerical values for following parameters: MOS, Average Throughput, Average RTT, Minimum RTT, Average Jitter, Packet Loss and RCA.

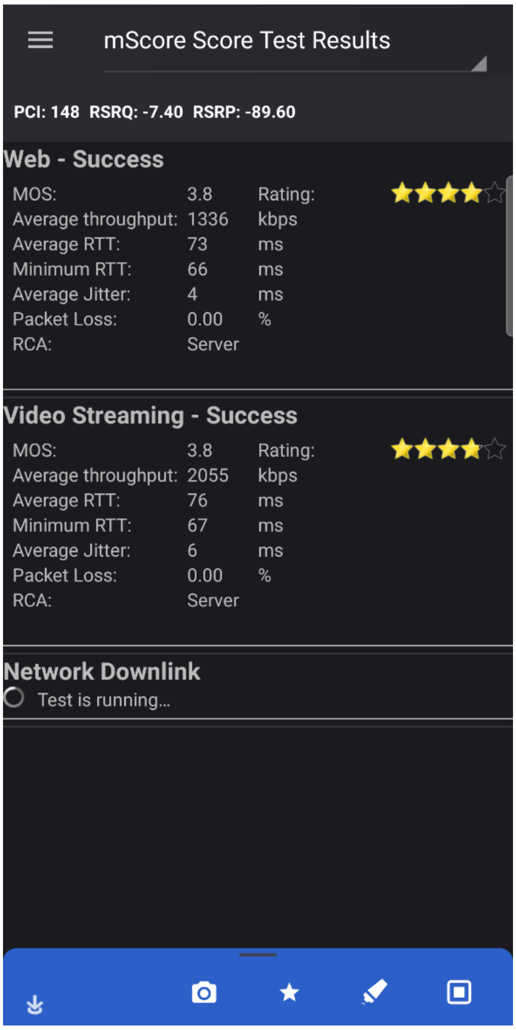


Figure 31. mScore Score Test Result page in Nemo Handy-A.

4.4.3 Custom Pages

Nemo Handy-A has a built-in page editor that allows the user to create new data pages. Custom pages are saved in a separate XML file, similar to the main display configuration file. The page editor supports page layouts without graphs, with simple horizontal bar graphs or with single or dual line graphs.

The page editor supports adding multiple sub-pages, which the user can navigate by swiping vertically. Each page may have a different layout and parameters. Each page can also be configured to be always visible, or visible only in a specific mobile system, such as GSMA, WCDMA or LTE.

4.4.4 Serving Info

Nemo Handy-A has a Serving Info view that, once enabled, is always visible on top of the display (see Figure 31). The parameters on the Serving Info view can be selected from the settings. The parameters are represented in the view in textual form with parameter’s label and value separated by a colon. The size of the text can be increased or decreased, and the color can be selected from nine options. System specific parameters are shown in the Serving Info view only when they match the active system of the measurement terminal.

4.4.5 Notifications

Nemo Handy-A supports notifications that can be configured between four representation actions: audio, icon, pop-up and wearable. Audio notifications are played on the measurement terminal speaker, when the notification is triggered. Notification icon is drawn above all line graphs and on the route on a map. For pop-up notification, the name of the notification is displayed in a popup view, also known as a toast in Android [42]. A wearable notification is displayed on a wearable device that is connected to the measurement terminal.

Nemo Handy-A comes with preconfigured set of notifications. The user can also create custom notifications by selecting a parameter and configuring a trigger for it (see Figure 32). A notification can be configured to trigger if the value of the selected parameter changes, or if it goes over or under a set threshold value, or if it equals the set threshold value. Custom notifications can be configured with a custom text-to-speech audio. Notification icon cannot be changed in the UI.

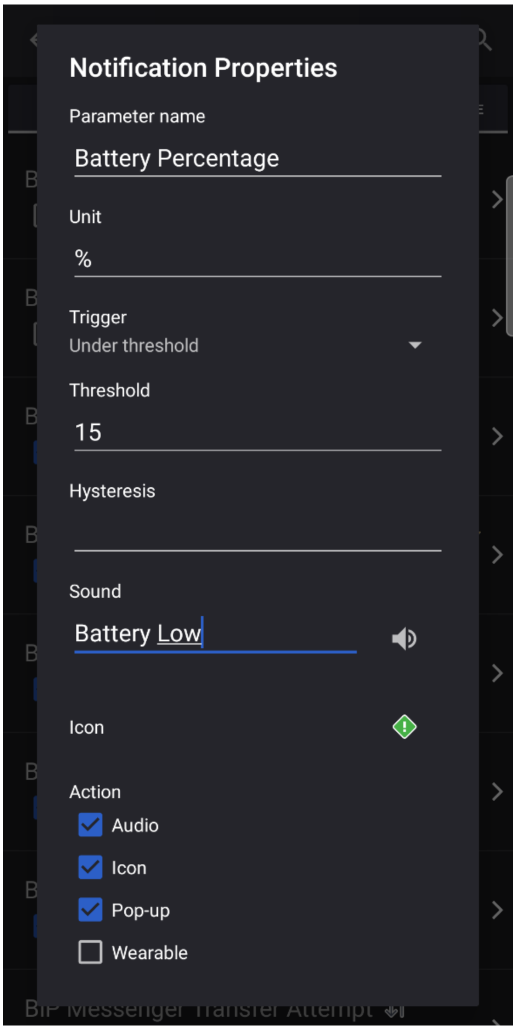


Figure 32. Notification Properties dialog in Nemo Handy-A.

5 Data Requirements

The analysis on this chapter is based on the quantitative research, described in section 2.2. Data quantity of the measurements can be analyzed using derived parameters: event rate and parameter rate. Event rate (λ_e) is obtained from expression:

$$\lambda_e = \frac{n}{t}, \quad (1)$$

where

λ_e is the event rate (1/s)

n is the number of measurement events

t is the duration (s) of the measurement.

Each measurement event consists of multiple related parameters. Some events have fixed number of parameters, while others dynamically repeat some parameters based on the file format and measurement conditions. Therefore, parameter rate is more accurate metric for the quantity of the data. It is obtained from expression:

$$\lambda_p = \frac{n}{t}, \quad (2)$$

where

λp is the parameter rate (1/s)

n is the total number of parameters in all measurement events

t is the duration (s) of the measurement.

Both event rate and parameter rate are calculated for all measurements in the dataset and individually for each event type within each measurement. Both data rate parameters can be used to specify requirements for different parts of the software. They can be used to determine the upper limit of how much data must be written into a database and into a log file. They also affect how many data points are drawn to data views, which is the focus of this research.

5.1 Data Quantity on Measurement Files

As discussed earlier, the dataset contains measurements from two different platforms with two different DUT types. Distribution of parameter rates are presented in Figure 33 for each platform and DUT type using a boxplot. The boxplot represents the distribution of the data between the first and the third quartile using a box, where the bolded line represents the median value. The whiskers next to the box represent the minimum and maximum values, except when the distribution of the data contains outliers that are further than one and a half times the box length away from the box edge. In this case the whiskers are cut, and the remaining outliers are drawn as small circles.

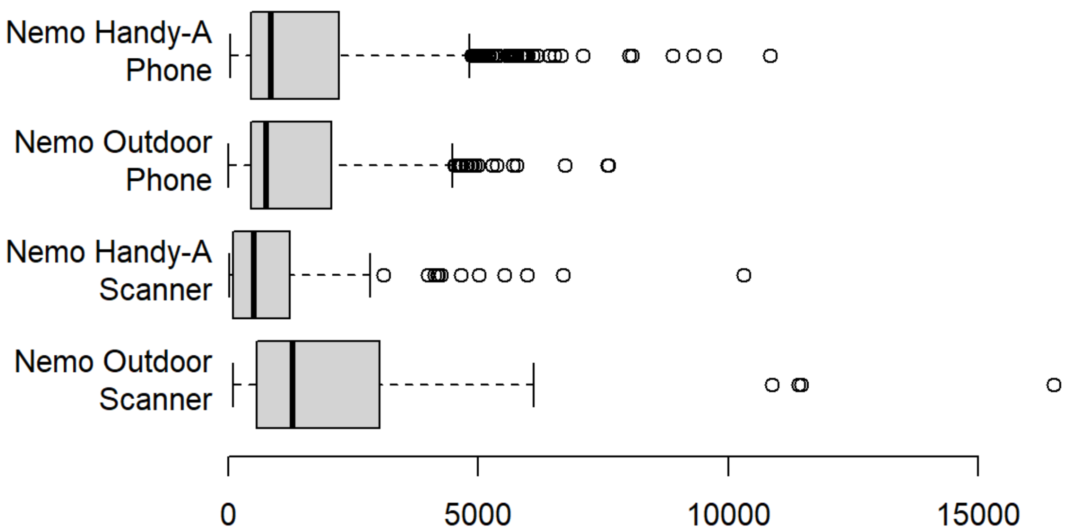


Figure 33. Parameter rate by measurement type.

Phone measurements for both Nemo Outdoor and Nemo Handy-A platforms have similar distribution, which can be explained with the use of Nemo Active Testing Application (NATA). Nemo Outdoor communicates with NATA, which is an application running on the Android phone that executes the transactions and collects the data. It is based on the same codebase as Nemo Handy-A, inherently producing similar data. The visual comparison also shows that network scanners used with Nemo Outdoor have the highest parameter rates.

One hypothesis is that more data is getting collected as the tools and network technologies evolve. This can be examined by comparing the parameter rates year-by-year over the five-year period of the dataset. In the Figure 34 this is done for all measurements in the dataset, where the DUT is a phone. While the first year in the dataset (2019) seems to have lowest parameter rates, there’s no clear upward trend visible for the years 2020 to 2023. This would indicate that parameter rates are not increasing at any significant rate. It is also possible that the rollout of NR networks is not yet clearly visible in the data and comparing measurements made in LTE and NR networks might yield different results. Such a comparison is not possible with the dataset of this research as it doesn’t contain information about the network technologies used in the measurements.

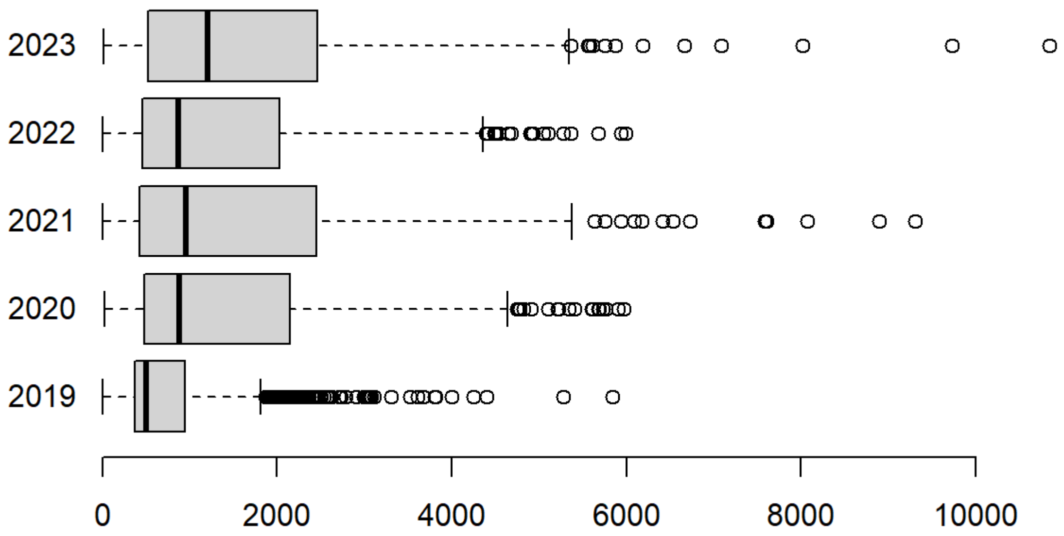


Figure 34. Parameter rate of phone measurements by measurement year.

When specifying the requirements for the system, maximum data rates of each unique measurement event can be examined. Maximum measurement specific parameter rate for each measurement event type is compared to their event rates in a scatter plot in Figure 35 for phone measurements. It contains 175 measurement events, of which 154 have less than 10 events per second and less than 220 parameters per second. Additional 13 measurement events have under 35 events per second and under 560 parameters per second.

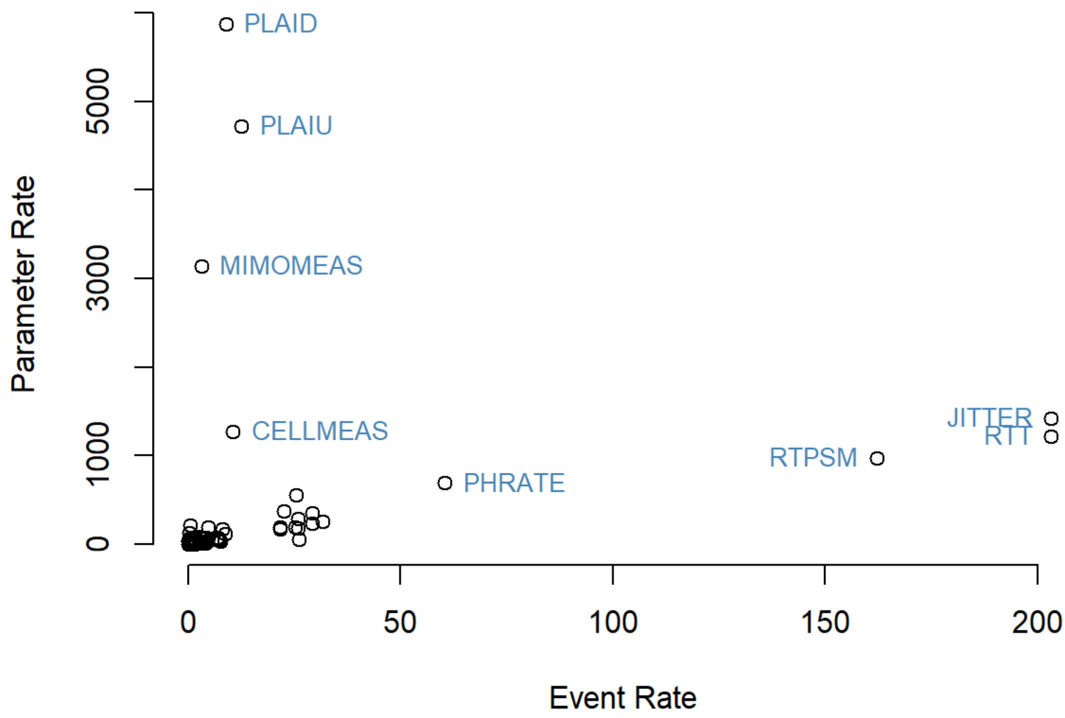


Figure 35. Highest data rate events for phone measurements.

Eight outliers can be identified with greater parameter rate or event rate than 95% of the event types. Measurement event types of JITTER, RTT and RTPSM reach over 160 events per second but with constant event size, their parameter rate stays below 1500 parameters per second. Conversely. measurement event types of PLAID, PLAIU and MIMOMEAS have lower event rate, but at maximum over 1500 parameters per measurement event.

For scanner measurements the total number of unique measurement event types is 19, of which seven have maximum parameter rate over 100 parameters per second. These measurements with their maximum parameter rates are plotted in Figure 36 against their corresponding event rates. Measurement event types of CELLSCAN, FREQSCAN, OFDMSCAN and SPECTRUMSCAN have over 600 parameters per measurement event.

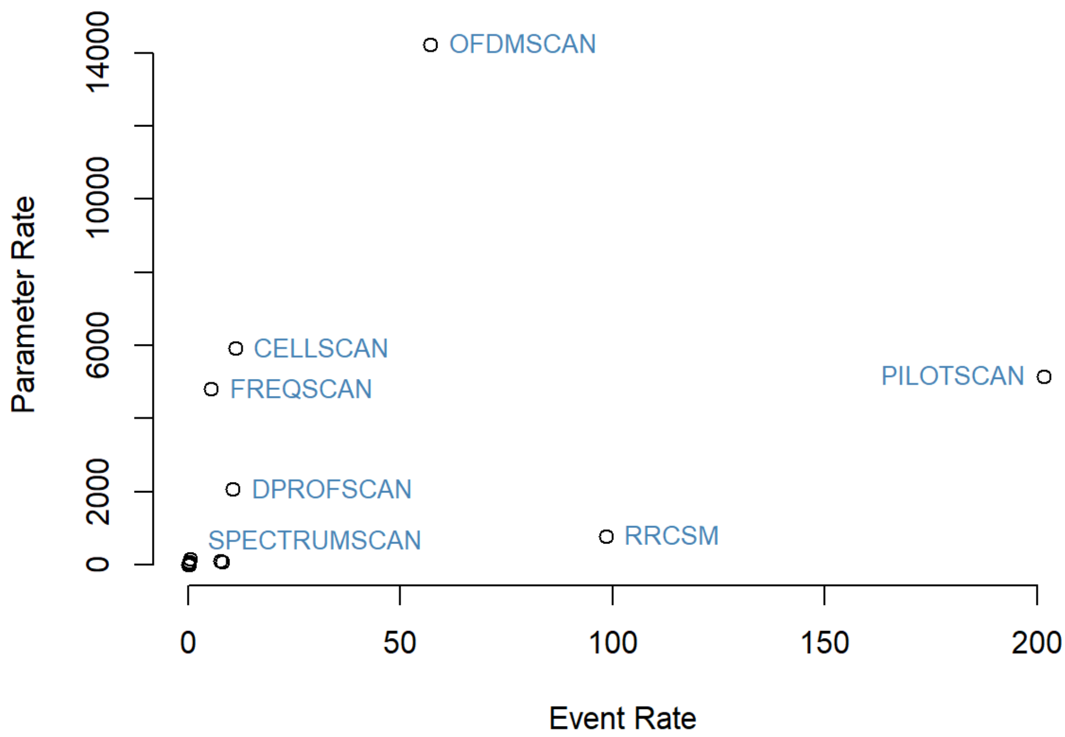


Figure 36. Highest data rate events for scanner measurements.

The dataset contained only two measurements with SPECTRUMSCAN events. Both measurements had only a brief period of spectrum scanning of the total duration of the measurement, decreasing the calculated event and parameter rates. If the measurements plotted above with SPECTRUMSCAN event rate of 0.41 events per second and 170 parameters per second would be cut to contain only the spectrum scan measurements, they would have event rate and parameter rate of 16 $^1/_s$ and 6700$^1/_s$, respectively. The measurement with highest data rates on SPECTRUMSCAN events is analyzed further on chapter 5.2.3.

The measurement event types identified in Figure 35 and Figure 36 are taken to closer analysis. The Figure 37 presents the distribution parameter rates of each of these measurement events on the dataset, sorted by their median value. The scale is logarithmic as the maximum value is 14000 parameters per second, while 90% of the data points are below 520 parameters per second, median value being 44 parameters per second.

In the lower end, the first quarter of the data points have fewer than 12 parameters per second. Corresponding event rates for the first quarter are below 0.8 events per second. Measurement events like RTT are written by transactions that may not be executed throughout the measurement period, reducing the overall data rates when examined over the measurement length.

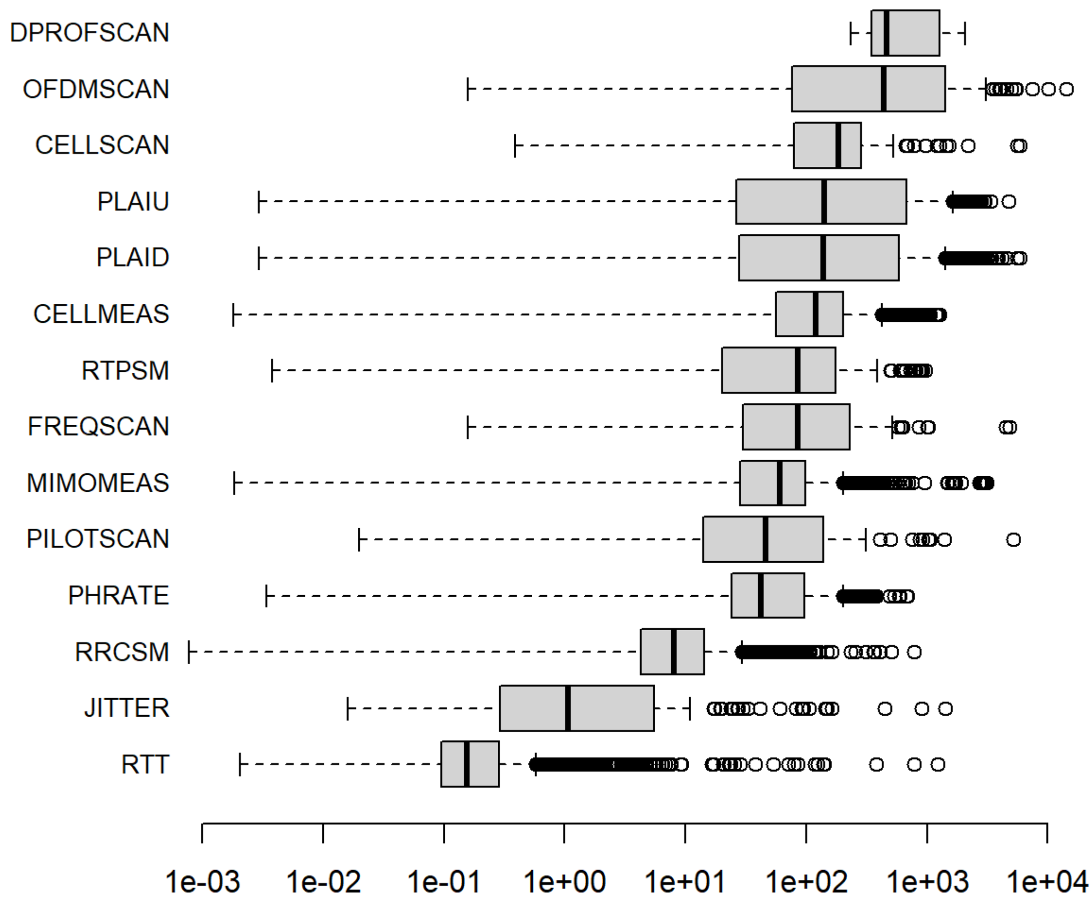


Figure 37. Parameter rate distribution of measurement evets.

The Figure 37 shows that the data rates presented in Figure 35 and Figure 36 are outliers. DPROFSCAN doesn't show outliers as it is based only on three data points. Other measurement event types are found from at least 66 measurements, and half of the listed measurement event types are found from over 2000 measurements each. The outliers are still worth closer analysis. If their data in those measurements is valid, they set the requirements for the maximum data rates. This study looks at some of those measurements and measurement events using Nemo Outdoor and Nemo Handy-A in the following chapter.

5.2 Data Quantity on Graphs

A single graph or a data grid may contain data from a single measurement event, or multiple measurement events. Depending on the configuration of the data view, only a set of the total parameters from a measurement event is displayed. Graphs potentially pack more data into a single data view than data grids, which rely on textual presentation. This chapter examines how the previously identified high data rate events are presented currently on Nemo collection tools.

5.2.1 Line Graph

Line graph is widely used and arguably the most important data visualization type for the time series data in the network measurement collection tools. For Nemo Handy-A, it is the most data dense visualization component. Figure 38 demonstrates how Reference Signal Received Quality (RSRQ) values in dB from the same measurement are drawn in a line graph on Nemo Handy-A and Nemo Outdoor. The measurement file has in average 12 CELLMEAS events in a second with the average of 62 parameter each. The CELLMEAS event contains cell measurement parameters, including RSRQ values for a primary sell and detected cells. The measurement used in this example has typically around four detected cells, but not any secondary cells.

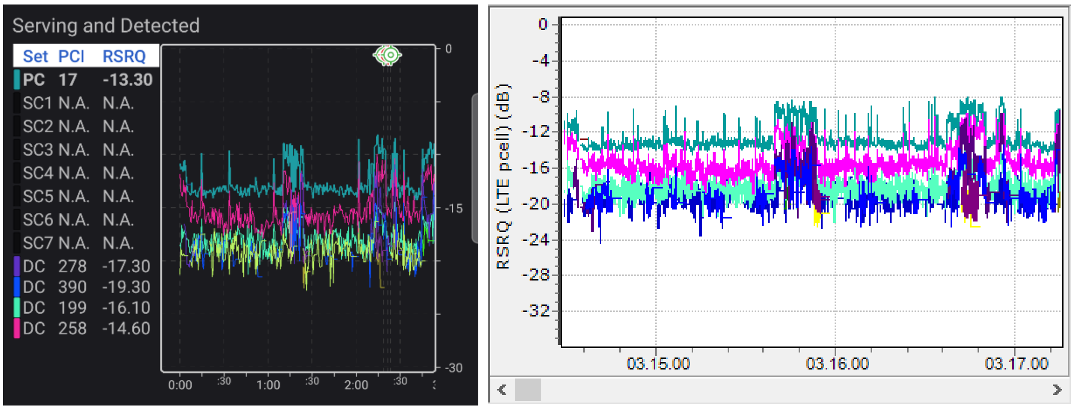


Figure 38. Line graph in Nemo Handy-A (left) and Nemo Outdoor (right).

Nemo Handy-A has a fixed scale on the x-axis. It also uses down sampling by dropping values, if the time between events is too short. This can be seen in the figure above, as the line graph in Nemo Handy-A doesn't have as many spikes on the primary cell (PC or pcell) line as it does on the line graph on Nemo Outdoor. The overall shape and trend of the values is still visible in both line graphs, even the Nemo Handy-A presents only about one sixth of the data points compared to Nemo Outdoor on this example.

The number of lines in the line graph also effects on how much data is represented in the graph component. In the example above, the graph will draw a line for each primary, secondary and detected cell value that are available in the measurement. With multiple lines, the graph becomes harder to read and already with average of five lines, the rendering order of the lines changes how the graph looks as some of the lines become hidden behind the lines drawn last.

5.2.2 Scatter Graph

Scatter graph is used to represent the relationship between two parameters. The values accumulate over time, which causes the data points to be overlayed. This is demonstrated in Figure 39, using the same measurement file from the previous example. The scatter graph plots primary cell RSRQ and RSRP values with average of 12 data points added every second.

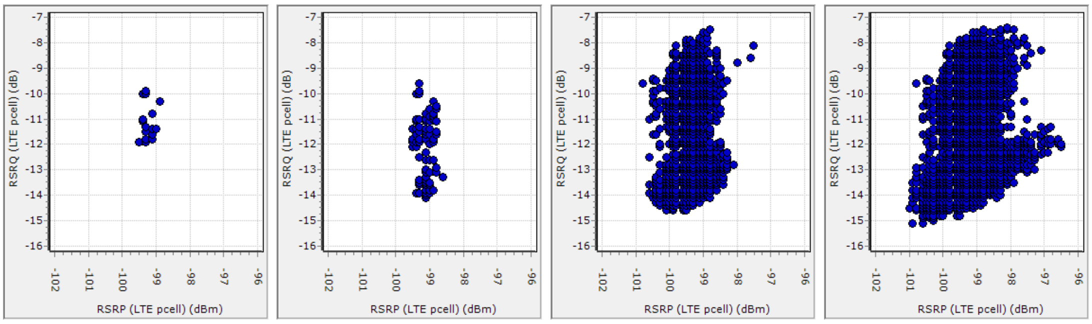


Figure 39. Scatter graph (RSRQ versus RSRP) on Nemo Outdoor at 1 second, 10 seconds, 1 minute and 1 hour into the measurement.

The Scatter Graph component in Nemo Outdoor represents all the data points from the beginning of the measurement. It does not have a reset functionality like the peak line on the Spectrum Graph component. It also does not have a way to limit the data points to the most recent ones.

To avoid the overplotting issue in a scatter graph, the same data could be represented with a 2D histogram. One common variant of a 2D histogram is a hexbin plot, that divides the graph area into hexagon shaped bins that are colored based on the number of data points in its area. Using the same data from the previous example at 1 hour mark, the hexbin plot in Figure 40 represents the distribution of the data with color. The plot is generated by exporting the data from Nemo Outdoor Event Grid into a CSV file, importing and processing it with an R-script and plotting using ggplot-library.

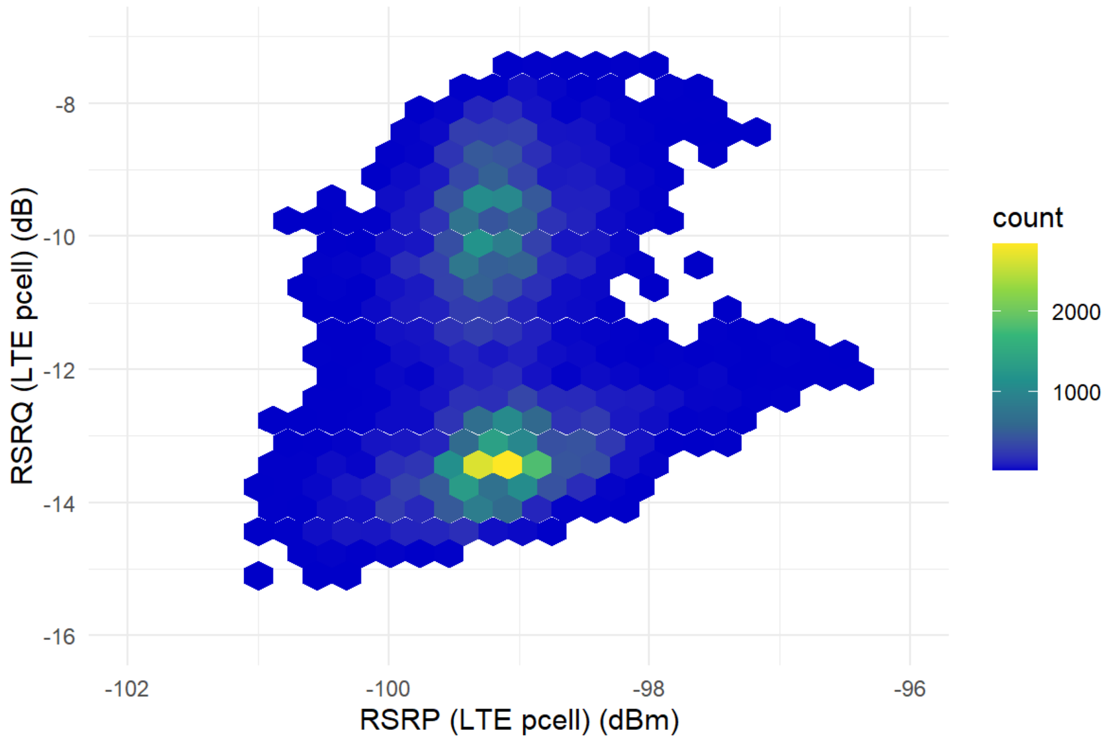


Figure 40. Hexbin plot (RSRQ versus RSRP) generated with ggplot.

5.2.3 Color Grid Graph

Spectrum scan measurements are typically represented in Nemo Outdoor and other tools using a spectrum graph that represents the signal level in y-axis and the frequency range in x-axis. The same data can be represented in time domain by having axes for time and frequency, while representing the signal level with a color. This is another variant of 2D histogram, and it is often called a waterfall plot, when it's placed under the spectrum graph, sharing the x-axis for frequency range and representing the time domain vertically.

In Nemo Outdoor, the data window that can represent spectrum scan data in time domain is called a Color Grid Graph. It has the time domain horizontally like the Line Graph component. The measurement with highest SPECTRUMSCAN parameter density in the data set is represented in a Nemo Outdoor Color Grid Graph in Figure 41.

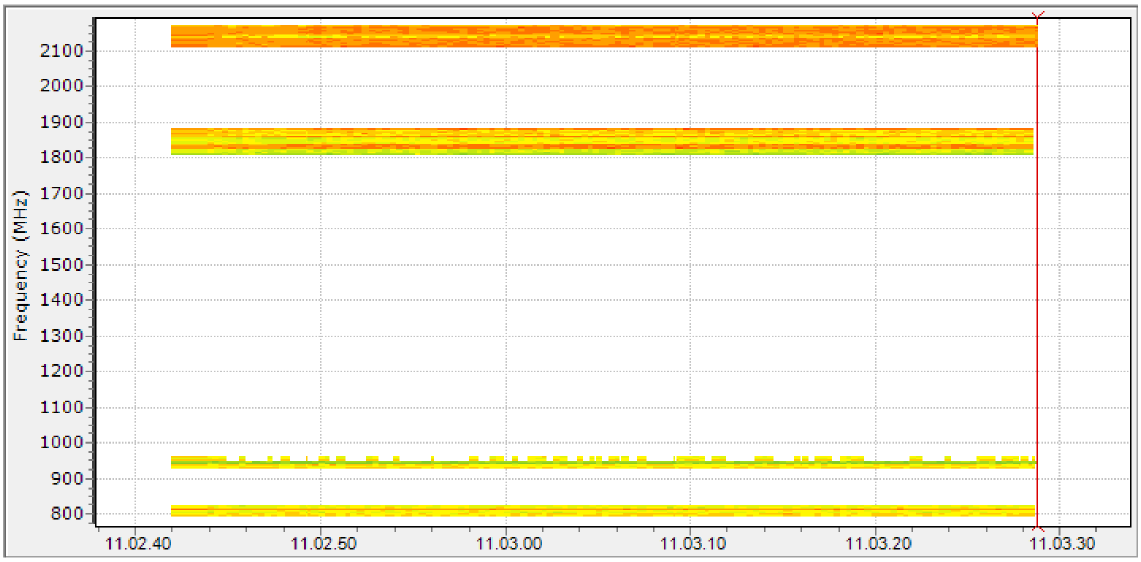


Figure 41. Spectrum scan measurement in a Color Grid Graph in Nemo Outdoor.

The measurement configuration has four frequency bands, which are scanned for 45 seconds. In this measurement period, the TSMA6 scanner has recorded over 3000 data points with frequency and RX level values. Drawing all the values in the Color Grid Graph is a heavy operation and it decreases the responsiveness of the Nemo Outdoor user interface.

6 User Requirements for Data Representation

The analysis in this chapter is based on the interviews conducted as a part of the qualitative research, described in section 2.3 and is an implementation of requirements analysis process as described in section 3.3. The interviewees are labeled as described in methodology section 2.3.

It is crucial to acknowledge that network measurement tools are used for various use cases (see section 4.2), each having unique set of requirements. The form factor of smartphone-based Nemo Handy-A and PC-based Nemo Outdoor also effect on their typical use cases and requirements.

6.1 Value of Network Measurement Data

Network measurement data is utilized in several ways, including optimizing a network, verifying new features and software, troubleshooting network issues or benchmarking networks of multiple operators against each other.

Throughput rates are found as one of the most important key performance indicators (KPIs) for data testing by both interviewed customers. For voice testing, Operator_3 highlights the value of call drop events and failed call attempts. Other highlighted KPIs are handover failures, signal levels and success rates of applications being tested, MOS score for voice quality testing, packet technology, and GPS information.

Often the more profound value is not in a single parameter, rather than the correlation between multiple parameters, as stated by Nemo_4. As an example, Consultant_2 mentions that the expected throughput of a YouTube test is much greater than one of a Facebook test. Operator_3 similarly notes that expected throughput in NR system is greater than in LTE. Therefore, looking at just the throughput value does not provide all the value. They continue that it’s important to distinguish between the throughput rates of each system if they are active simultaneously. In addition, Operator_3 describes how checking RSRP might explain a low throughput rate or indicate that further investigation is needed.

Nemo_2 claims that for troubleshooting cases, the software's value is in the ability to bring out the required set of parameters, from the hundreds that are available, to form an insight to the problem being investigated. While this can be accomplished using dedicated analysis and post processing software, Nemo_2 argues that for a single measurement, Nemo Outdoor provides a powerful UI that is not shy of the capabilities of those tools. Operator_1 agrees, pointing out the familiarity with the customized workspace they have created for Nemo Outdoor. For multiple measurements, such as a benchmarking scenario, Consultant_1 is not expecting the collection tools to provide the analysis capabilities required for their use cases.

6.2 Use Case Aware Views

Possibly the most important theme that emerged on all the interviews was the concept of use case specific views. Each stakeholder had individual ideas and opinions around the theme, the consensus being that the network measurement tool's UI shall present the parameters that are the most relevant to the test that is being conducted.

Both customers have already implemented this concept by customizing workspaces in Nemo Outdoor for their specific needs with distinct approaches. While Operator_1 relies on a single custom workspace with 19 view groups for all their use cases, Consultant_2 reuse smaller custom workspaces for each type of measurement use case. Additionally, Nemo_4 proposed that the UI should dynamically adapt to the script that is running. For example, when a ping testing is running, the UI should show the RTT parameter, which is currently difficult to locate in Nemo Handy-A using the default display configuration.

Overall, all participants found that Nemo Handy-A would benefit more from the use case aware views, given its current limitations with the configurability and cluttered default display configuration with large number of pages. As Operator_2 describes:

> The user interface, in my opinion, is too cluttered and there's too much to swipe through both horizontally and vertically, lots of menus. And maybe it's too difficult to configure it to your personal needs. I've never got myself around to do it, as I use it only occasionally, it has been too difficult to customize the UI.

Operator_3 proposed that the software may provide multiple modes, including Basic, Advanced and Custom, where the simplest basic mode could cover 80% of the use cases. In Nemo_4's vision, the UI would be made of dashboards that allow drilling down to more specific views.

Configuration in Nemo Outdoor is better and new parameters can be easily pulled into the view as mentioned by Nemo_4. Outdoor also comes bundled with some workspaces that serve some use cases.

6.3 Operational Users and Automation

One theme that was brought up especially by Nemo_2, was the increase of operational users, which do not have expertise in the network technology domain. These users conduct the drive tests, indoor measurements, or other network measurement scenarios without understanding the data. This reduces the importance detailed data representation, while simultaneously increasing the importance of easily validating the integrity of the data as discussed below in section 6.4.

Another related trend is the increasing level of automation, which is manifested in different ways on various markets. According to Nemo_2, the European interpretation of automation is to reduce the work done by humans, while in some other regions it might mean the capability to monitor and control the progress of the operational user remotely in real time, effectively employing two individuals for the task. In both cases, the need for data representation is moved away from the measuring unit. Taken further, the automation could include the use of artificial intelligence and machine learning, which would be a game changer according to Nemo_4. This would also influence the data representation requirements, as the data should be formatted for the algorithm and not for the human to find insights from. Only the results of that automated analysis would be represented to human audience.

6.4 Data Integrity

Many network measurement use cases depend on post processing the data, which shifts the need for data representation to an analysis tool or a report that is the final product of the measurement. In these cases, the primary focus during the measurement is to validate the integrity of the data being collected as described by Nemo_4. On the same point, Consultant_1 describes their typical benchmarking drive test setup:

> What we want to make sure during the measurements is that the data we collect, it's also valid for the analysis. And for that purpose, we typically have a test engineer inside the car that is basically all the time monitoring that the measurement is running smoothly and also that what we get is valid for the analysis.

There could be multiple reasons why the data might not be valid for the analysis. As an example, Operator_1 describes a case, where the script was configured to run voice quality testing on LTE system, expected result being a measurement file full of VoLTE calls with MOS values for voice quality. During a long drive test, the measurement terminal had dropped to 3G system and for an unknown reason it never recovered back to LTE, resulting the script to wait for the system change and never executing the voice calls and therefore the measurement was lacking the expected MOS scores. Operator_1 also notes that the stability and reliability of Nemo tools have been improving over the years.

Discussed solutions for validating the data integrity were mostly centered about the use case specific data views, discussed above in section 6.2. Notifications were also identified as a possible tool to help identify the issues during a measurement, as discussed below in section 6.5.3.

6.5 View Components

All participants found the current set of view components to be sufficient and could not name a new type of graph that they would need. Nemo_3 agreed that some heatmap type graph would be nice, as the idea was suggested by the author. Some areas of improvement were identified with the features and default configuration of current view components as discussed below.

6.5.1 Graphs

Line graph was rated as the most important graph type by Operator_1. Consultant_2 requested that the line graphs for a throughput parameter should differentiate between the active transactions. Additionally, they requested a filtering functionality in the line graph for different test types. Nemo_3 felt that in Nemo Handy-A, the default line width was too narrow and the scales too small to read. Nemo_4 also mentioned that some line graphs in Nemo Handy-A get very cluttered as too many lines are drawn in one graph and that it is difficult to match the colors with the legend as they are not distinct enough.

Configuration of graph components was found unintuitive by Operator_2, who compares them to Excel and other visualization software that allows editing an axis by double-clicking, whereas in Nemo Outdoor the configuration is always done through menus.

Gauge view was called as “marketing material” by Nemo_3, and according to Operator_1 it is only useful, when presenting the results to a non-expert audience.

Possible improvements for existing graph components were discussed during the interviews, including adding a support for spectrum graph type peak lines for other graph types. Another topic was the ability to group or stack bars in bar graph. While the participants felt that these might be good additions, they did not find an immediate practical need for these features.

6.5.2 Serving Info

Serving Info component in Nemo Handy-A is presented in section 4.4.4. It's importance in data representation was recognized by many of the participants, although Operator_1 admits they rarely use it. Even with its limitations, Nemo_4 argues that it "serves the purpose" and Nemo_3 would enable it by default. Nemo_5 finds it especially useful for indoor measurements, where most of the UI is preserved for the indoor map and user's focus is adding markers on it.

Nemo_1 would like to improve the way Serving Info presents data of multiple technologies, for cases like NR NSA, where it would be important to see both LTE and NR parameters. They also mention that the Serving Info would be easier to read, if each parameter would have a dedicated location on it. Nemo_5 would like to expand the functionality of Serving Info even further by adding graphs and other visualization capabilities.

6.5.3 Notifications

Notifications, as presented in sections 4.3.5 and 4.4.5, are distinct from other data representation methods as they can be represented visually on top of other data views and also audibly. Operator_1 uses notifications mostly for dropped voice calls, while Consultant_1 considers their value for ensuring that the measurement is running as expected and producing valid data for the analysis, especially if the most important KPIs cannot be monitored on a single screen.

Consultant_1 suggested that notifications could be improved by adding a condition that takes the timeline into account. Then the notification would trigger for example, if the data rate has been under a certain threshold for fifteen seconds, or if the measurement terminal has been on unwanted system for too long. They find that it is important to manage the number of notifications that keep popping up, to bring user's attention to important events.

Operator_1 was also open to the idea of a smarter notification that would notify the user if a transaction specific expected results failed to occur. As an example, if a voice call transaction could be configured with a notification that expects MOS events to occur during the call, it would help the user to detect situations, where they are missing.

6.6 Nonfunctional requirements

Configurability is defined as a key requirement by Nemo_2. Nemo_5 also finds configurability desired, while acknowledging current architecture’s limitations. All the participants agree that the data views in Nemo Outdoor are way easier to configure than ones in Nemo Handy-A. Ability to configure Nemo Handy-A views on a PC was requested by Nemo_1 and Operator_2.

Many participants also find that due to its form factor and typical use cases on the field, Nemo Handy-A has more to improve also in terms of readability. Nemo_3 and Nemo_4 requested a larger font for the graph scales, while Operator_1 and Operator_3 were hoping for an easy-to-read view that would contain the key information with a large font that would be easy to glance at during a drive test.

Performance issues were noted only by Nemo_1 regarding the Color Grid Graph in Nemo Outdoor.

6.7 Nemo Outdoor and Nemo Handy-A

Both interviewed customers use both Nemo Outdoor and Nemo Handy-A. The product is selected by the use case, or occasionally both products are taken for a measurement campaign to achieve higher number of measurement terminals as described by Operator_1. As the mixed use of these products is common, Nemo_4 proposes that they should have same look and feel. Sharing the configuration files between the products would help with the configurability discussed above.

Most of the critique and requirements were targeted to Nemo Handy-A, which is more challenging in terms of data representation due to its form factor. Operator_1 describes the benefits of Outdoor as follows: “I personally like that in Outdoor, you’re able to visualize much more data onto the big screen, which is why I mainly use it.”

Nemo Handy-A was found better suited for indoor measurements by all the participants. Nemo_1 also highlighted its WiFi Spectrum Graph, that is not available in Nemo Outdoor. On Nemo Handy-A’s limitations that were not covered on above sections, Nemo_4 mentions the inconsistency on naming some of the parameters compared to Nemo Outdoor and 3GPP specification, as well as the limited time window of line graphs without panning or zooming capabilities.

7 Result Summary

The summarized answers to the research questions are provided below in reverse order, starting from the third sub question and ending to the main research question.

7.1 What Type and Quantity of Data do Network Measurement Tools Record in Nemo File Format?

The type and quantity of data depends on multiple factors, including DUT type, network environment and the test type. Scanners typically produce more data than phones. The data visualized on the collection tool may contain thousands of data points per data view (see section 5.2).

The high data rate measurements identified in the quantitative research component of this study (see chapter 2.2) can be used to further develop the data requirements for the data collection tools, including their architecture and UI components.

7.2 What Data Representation Limitations do Nemo Collection Tools Have?

No need for new graph types were identified. The limitations were mostly on usability, more precisely on configurability and lack of use case specific views. Nemo Handy-A was found more limited in these areas compared to Nemo Outdoor.

7.3 Where is the Value in Network Measurement Data for the Users?

The value of the network measurement data depends on the use case. For a benchmarking scenario, it might be packed in few main KPIs like throughput, application success rates and MOS scores. For an optimization or verification use cases, the value is typically in the correlation of multiple parameters. For troubleshooting, the value is even deeper in the details of the data.

For many use cases, the value of the data is discovered in post processing, and it might be represented in a form of a report.

7.4 What are the Data Representation Related User Requirements on Network Measurement Tools?

The data representation related user requirements are mostly satisfied by Nemo collection tools. The major enhancement requirement is a better support for various use cases in the network measurement field. The details need to be specified on future work.

Better use case support may be achieved by improving the default display configurations and their configurability. The overall user experience (UX) and workflow could be rethought to better suit the needs of various use cases. One possibility is also to automatically adapt the UI based on the test that is being conducted.

8 Conclusion

This work was conducted as a part of an iterative requirements engineering process for Nemo Outdoor and Nemo Handy-A to elicit and analyze user requirements for data representation on network measurement applications. Requirement engineering techniques of UI analysis, data analysis and interviews were used to respectively carry out the comparative, quantitative and qualitative parts of this mixed-method research.

Due to the changes in the Nemo organization that occurred during this research project, the research approach had to be adjusted and is not as user centric as originally planned. Only two customers with five individual users were involved in the requirements elicitation via interviews. This change consequently increased the significance of the comparative and quantitative components of the work. As such, this work can be considered as a single iteration of the requirement elicitation and analysis.

The main finding of the research is that the user requirements on various network measurement use cases are different and would be best satisfied if the measurement tools would adapt to the use case. The adaptation to various use cases could be achieved by offering a comprehensive set of tailored view configurations, by UI that dynamically adapts to the tests being conducted, by improved customizability, or ideally by some combination of all these solutions.

Requirements engineering process for Nemo collection tools may be continued using the results of this work. Finally, the developed requirements may be used in the enhancement projects of the associated software, when further developing the data architecture and designing more usable UI. Applicability to other network measurement software or solutions might be limited due to this work’s focus on Nemo collection tools.

It would be beneficial to include more users with various requirements to develop more accurate user requirements for the data representation on network measurement tools. Other requirement elicitation techniques, such as observation, could bring more insight into the needs of the end users. Further iterations of the requirements development process could also utilize prototypes of use case specific data views, smarter notifications, and other enhancements on the current functionality.

The data analysis from the quantitative component of this work could be improved by parsing and analyzing the data against the Nemo File Format specification. This work would be beneficial when improving the data architecture of the software and adjusting the data views.

Future research could focus on finding the use cases of network measurement tools and their unique requirements. Other research approaches could be evaluation of graphical component libraries and analysis of other network measurement products. Future work could also focus on improving the identified usability issues, especially on Nemo Handy-A, or find new ways how the UI and workflow on a mobile network measurement application could be designed for better user experience.

References

[1] The 3rd Generation Partnership Project, “Introducing 3GPP.” Accessed: Apr. 29, 2024. [Online]. Available: https://www.3gpp.org/about-us/introducing-3gpp

[2] M. Rouse, “Base Transceiver Station,” Techopedia. Accessed: Feb. 17, 2024. [Online]. Available: https://www.techopedia.com/definition/2927/base-transceiver-station-bts

[3] International Telecommunication Union, “ITU-T Y.2060,” Jun. 2012. Accessed: Apr. 22, 2024. [Online]. Available: https://www.itu.int/rec/T-REC-Y.2060-201206-I

[4] P. Bourque and R. E. (Dick) Fairley, Eds., “Software Engineering. Guide to the software engineering body of knowledge (SWEBOK),” IEEE Computer Society, London, 2016. doi: 10.3403/30314312.

[5] Google Developers, “KML Tutorial.” Accessed: Jan. 17, 2024. [Online]. Available: https://developers.google.com/kml/documentation/kml_tut

[6] International Telecommunication Union, “ITU-T P.800.1,” Geneva, Jul. 2016. Accessed: Apr. 22, 2024. [Online]. Available: https://www.itu.int/rec/T-REC-P.800.1-201607-I

[7] 3rd Generation Partnership Project, “3GPP TS 36.214,” vol. 18.0.0. 3rd Generation Partnership Project (3GPP), 2024. Accessed: Apr. 22, 2024. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/36_series/36.214/36214-i00.zip

[8] 3rd Generation Partnership Project, “3GPP TS 38.215,” vol. 18.2.0. 3rd Generation Partnership Project, 2024. Accessed: Apr. 22, 2024. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.215/38215-i20.zip

[9] Inc. Object Management Group®, “WHAT IS UML.” Accessed: Mar. 28, 2024. [Online]. Available: https://www.uml.org/what-is-uml.htm

[10] L. Quin, “Extensible Markup Language (XML),” World Wide Web Consortium (W3C). Accessed: May 02, 2024. [Online]. Available: https://www.w3.org/XML/

[11] International Telecommunication Union, “Strategies and policies for the deployment of broadband in developing countries,” Geneva, 2021. Accessed: Aug. 08, 2024. [Online]. Available: https://www.itu.int/hub/publication/d-stg-sg01-01-2-2021/

[12] M. Reglitz, “The socio-economic argument for the human right to internet access,” *Polit Philos Econ*, vol. 22, no. 4, 2023, doi: 10.1177/1470594X231167597.

[13] Organisation for Economic Co-operation and Development (OECD), “OECD broadband statistics update.” Accessed: Aug. 08, 2024. [Online]. Available: https://www.oecd.org/sti/broadband/broadband-statistics-update.htm

[14] Ericsson, “Ericsson Mobility Report,” Stockholm, Nov. 2023. Accessed: Mar. 20, 2024. [Online]. Available: https://www.ericsson.com/4ae12c/assets/local/reports-papers/mobility-report/documents/2023/ericsson-mobility-report-november-2023.pdf

[15] Keysight Technologies, “Nemo Wireless Network Solutions.” Accessed: Mar. 18, 2024. [Online]. Available: https://www.keysight.com/us/en/products/nemo-wireless-network-solutions.html

[16] U. Goel, M. P. Wittie, K. C. Claffy, and A. Le, “Survey of End-to-End Mobile Network Measurement Testbeds, Tools, and Services,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 105–123, Jan. 2016, doi: 10.1109/COMST.2015.2485979.

[17] T. Jiang, Y. Hou, and J. Yang, “Literature Review on the Development of Visualization Studies (2012–2022),” in *The 3rd IEEE International Conference on Electronic Communications, Internet of Things and Big Data 2023*, Basel Switzerland: MDPI, Jul. 2023, p. 89. doi: 10.3390/engproc2023038089.

[18] P. Tiirikainen, “Visualization of log data,” Master’s degree, Lahti University of Applied Sciences, 2018. Accessed: Aug. 05, 2023. [Online]. Available: https://urn.fi/URN:NBN:fi:amk-201804064185

[19] K. Vainio, “Customer-oriented collaboration in requirements engineering : an approach for business and IT units to successfully agree on customer needs,” Master’s thesis, Turku University of Applied Sciences, 2017. Accessed: Aug. 05, 2023. [Online]. Available: https://urn.fi/URN:NBN:fi:amk-201705249902

[20] L. Jussila, “The User Interface as a Part of the Requirement Specification,” Master’s thesis, Laurea University of Applied Sciences, 2020. Accessed: Aug. 05, 2023. [Online]. Available: https://urn.fi/URN:NBN:fi:amk-2020122329933

[21] P. Pisilä, “Langattomien verkkojen mittaustulosten rakenne käytännön mittauksissa,” *unpublished*. Oulu, 2023.

[22] OpenAI, “ChatGPT.” Accessed: Apr. 21, 2024. [Online]. Available: https://openai.com/chatgpt

[23] K. E. Wiegers and J. Beatty, *Software Requirements*, 3rd ed. Microsoft Press, 2013.

[24] E. Hull, K. Jackson, and J. Dick, *Requirements Engineering*. London: Springer London, 2011. doi: 10.1007/978-1-84996-405-0.

[25] I. Sommerville, *Software Engineering, Global Edition*, 10th ed. Harlow: Pearson Education, Limited, 2016.

[26] J. Voas, “Protecting against what? The Achilles heel of information assurance,” *IEEE Softw*, vol. 16, no. 1, pp. 28–29, 1999, doi: 10.1109/52.744563.

[27] P. Bourque, R. Dupuis, A. Abran, J. W. Moore, and L. Tripp, “The guide to the Software Engineering Body of Knowledge,” *IEEE Softw*, vol. 16, no. 6, pp. 35–44, 1999, doi: 10.1109/52.805471.

[28] P. Lundmark, “Thirty years on from the call that transformed how we communicate.” Accessed: Mar. 18, 2024. [Online]. Available: https://www.nokia.com/blog/thirty-years-on-from-the-call-that-transformed-how-we-communicate/

[29] J. Tuomivaara, “Nemon tuotteiden syntyhistoriaa,” *unpublished*.

[30] Elektrobit Group Plc., “THE SALE OF ELEKTROBIT’S NETWORK TEST BUSINESS TO ANITE HAS BEEN CLOSED.” Accessed: Aug. 13, 2023. [Online]. Available: https://www.bittium.com/index.php?id=1615&locate=200611%2F1091006

[31] Keysight Technologies, “Keysight Technologies Completes Acquisition of Anite.” Accessed: Aug. 13, 2023. [Online]. Available: https://about.keysight.com/en/newsroom/pr/2015/13aug-gp15118.shtml

[32] Keysight Technologies, “Nemo Wireless Solutions.” Accessed: Aug. 13, 2023. [Online]. Available: https://www.keysight.com/us/en/solutions/5g/nemo-wireless-network-solutions.html

[33] Keysight Technologies Finland Ltd, “Nemo Handy User Guide.” 2024.

[34] Nemo Technologies, “NMS/X 5.2 Product Description.” 1998.

[35] Nemo Technologies, “TOM User Manual.” 1998.

[36] Nemo Technologies, “NIB Solo User Manual.” 2002.

[37] Nemo Technologies, “Nemo Outdoor User Manual,” 2003.

[38] Nemo Technologies, “Nemo Handy Display Guide.” 2006.

[39] Inc. Keysight Technologies, “Nemo Outdoor User Guide.” Keysight Technologies Finland, Ltd., 2024.

[40] Nemo Technologies, “Nemo Handy-W Display Guide,” 2010.

[41] H. Wickham, “Tidy Data,” *J Stat Softw*, vol. 59, no. 10, 2014, doi: 10.18637/jss.v059.i10.

[42] Android Developers, “Toasts overview.” Accessed: Jan. 27, 2024. [Online]. Available: https://developer.android.com/guide/topics/ui/notifiers/toasts

Comparative Data View Analysis of Nemo Outdoor and Nemo Handy-A

Table 3. Graph general features.

Feature	Outdoor	Handy
Number of graphs	Multiple per data window. Configured as groups in the Side Panel	0-2 per page
Zoom	Mouse scroll, or slider in the Toolbar: Line / Vertical Bar / Horizontal Bar / Spectrum / Color Grid Graph NOT in: Scatter / Spectrum Graph / Gauge Graph	Context menu (long tap over graph) in Vertical Bar Graph only
Other	Option to show/hide: Toolbar, Scrollbar, Axis, Side Panel, Layers, Values Side panel: Layers (groups), "Values" grid or "Extended value lists"	
Save as Image	Yes	No (Handy's generic screenshot functionality)
Device Selection	Device Selection (select device per parameter) Quick Device Selection (single device for all parameters)	Walker Air: Device selection per page Walker Air, Combined Line Graph: Always all Agents

Table 4. Line graph features.

Feature	**Outdoor**	**Handy**
X scale zoom	Yes	No
X scale scroll	Yes	No
Auto scroll	Yes, selectable	Always
Y scale position	Always on Left, can be hidden	Per parameter: None / Left / Right Graph can have y scales on both sides (for different parameters)
Y scaling	Per unit type in a group	Per parameter
Y automatic scaling	Yes Manual min & max	None / High / Low / Both Manual (min & max that are not automatically scaled)
Y threshold	Minimum and maximum Displayed as a horizontal line	No
Legend	Option to show on top of the graph, configurable transparency (Show only the first value of a list parameter)	Bottom Graph: Parameter labels and value above the graph Right Graph: Table left of the graph
Line color	Classic / Fixed Color / Color set / Algorithm	Bottom Graph: Fixed color Right Graph: Color from ValueKey
Line width	Configurable	Configurable
Drawn as area	Yes	No
Line as stairs	Configurable	Used for integers, not used for floating point values
Filters	Top-N Custom Filters	Bottom Graph: List index Right Graph: Max number of lines
Save as	Image	-

Table 5. Bar graph features.

Feature	Outdoor	Handy
Vertical bar graph	Yes	Yes
Horizontal bar graph	Yes	No

Table 6. Spectrum Graph.

Feature	Outdoor	Handy
Supported parameters	Configured in a configuration file	WiFi spectrum only
Markers	Yes	-
Peak Lines	Configurable width and color	-
Line Properties	Same as Line Graph	-

Table 7. Color Grid Graph.

Feature	Outdoor	Handy (N/A)
Standard graph features	Yes	-

Table 8. Gauge Graph.

Feature	Outdoor	Handy
Standard graph features	Yes	Yes

Table 9. Grid general features.

Feature	Outdoor	Handy
Font Size	Small / Default / Large / Extra Large	1-90
Text Style	-	Normal / Bold / Cursive
Font Color	Configurable per parameter: Black or white (contrast to the background) Color set	White (light grey)
Background color (Bar Graph)	Configurable per parameter (white by default): Color set	
Save as	Text / HTML file / Image / CSV File	N/A (Handy's generic screenshot functionality)
Copy	Text / Image	-
Multiselect	Shift-key Select All (context menu) or Ctrl+A	-

Table 10. Parameter grid features.

Feature	Outdoor	Handy
Data source	Multiple devices (columns)	Single data source, except scanner and IoT results are available with the host terminal data
Background color (Bar Graph)	Standard	Black Bar graph shown in a column right of the value, configurable with Min/Max/Limit values and Good/Bad colors
Presentation modes	Outdoor Preferences → Presentation	Handy Settings

Table 11. Table grid features.

Feature	Outdoor	Handy
Data source	Single device (parameters as columns)	Single data source (ValueDb), except scanner and IoT results are available with the host terminal data
Background color (Bar Graph)	Standard	Black Bar graph shown in a column right of the value, configurable with Min/Max/Limit values and Good/Bad colors
Filters	Grid Filters	

Table 12. Event grid features.

Feature	Outdoor	Handy (N/A)
Data source	Single device (parameters as columns)	
Notifications	Optional	
Save as	Standard + Decoded Text / MapInfo Tab-File	
Copy	Standard + Decoded Text	
Filters	Grid Filter Configuration, Quick Filter	
Decode Message	Double click / Automatic Decoded Layer 3 message or measurement event information	

Table 13. Packet grid features.

Feature	Outdoor	Handy (N/A)
Data source	Single device	
Save as	Text / HTML file / Decoded Text / Image	
Copy	Standard + Decoded Text	
Filters	Grid Filter Configuration, Quick Filter	
Decode Message	Double click / Automatic Decoded Frame	

Table 14. Trace grid features.

Feature	Outdoor	Handy (N/A)
Data source	Single device	
Save as	Text / HTML file / Decoded Text / Image	
Copy	Standard + Decoded Text	
Filters	Grid Filter Configuration, Quick Filter	
Decode Message	Double click / Automatic Trace Data Block	

Table 15. Sampling grid features.

Feature	Outdoor	Handy (N/A)
Data source	Single device	
Parameters	Multiple, in columns	
Sampling interval	Integer seconds	

Table 16. Map features.

Feature	Outdoor	Handy
Google Map	Yes	Yes
Open Street Map	Yes	Yes
MapXtreme	Yes	No
Indoor Floorplan over map	Only MapXtreme	Yes
Route	Multiple (with configurable offset) GPX only	Single route GPX or marker route
Font Size	Small / Default / Large / Extra Large	-
Save as	CSV File / Image	N/A (Handy's generic screenshot functionality)
Export to	Google Earth (*.kmz)	NemoNavi (navigate to selected .gpx file)
Base Stations	Yes	Yes
Rotate	No	Yes, using two fingers or compass. Tap to rotate to 0 degrees
Textual Comments	Yes	No

Table 17. Indoor map features.

Feature	Outdoor	Handy
Implementation	Indoor Window (MapXtreme) Indoor Map (Floorplan image)	"Map" page, over Google/OSM map
Route	Multiple (with configurable offset) Marker route only	Single route GPX or marker route
Font Size	Small / Default / Large / Extra Large	-
Save as	CSV File / Image / MapInfo Tab-File	N/A (Handy's generic screenshot functionality)
Base Stations	Yes	Yes
Rotate	No	Yes

Interview Support Slides

TURKU AMK
TURKU UNIVERSITY OF
APPLIED SCIENCES

Data Representation Requirements on Network Measurement Applications

Perttu Pisilä
2023

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

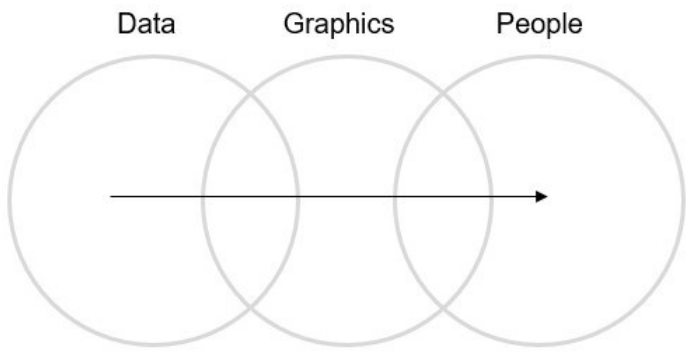
Author

- Perttu Pisilä
- Living in Oulu, Finland
- Working with Nemo Wireless Solutions
    - Trainee (2013 – 2014)
    - Customer Support Engineer (2014 – 2018)
    - Software Developer (2018 – present)
        - Nemo Handy-A and related Android software
- Bachelor of Engineering, Oulu University of Applied Sciences, 2014
- Studying at Turku University of Applied Sciences (2022 – present)
    - Master of Engineering, Software Engineering and ICT

TURKU AMK
TURKU UNIVERSITY OF
APPLIED SCIENCES

Research Questions

1. What type and quantity of data do network measurement tools record in Nemo File Format?
2. What data representation options shall network measurement tools support?
    1. Where is the value in network measurement data?
    2. What data representation limitations do Nemo collection tools have?
3. ~~Additional research question omitted from the appendix~~



TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

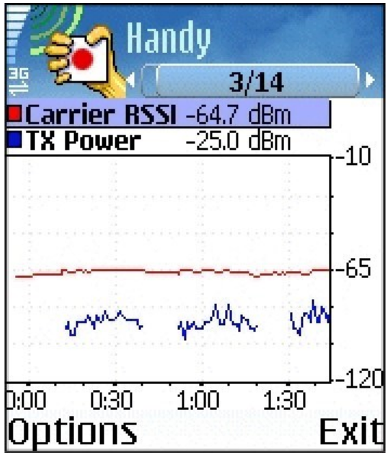
Major Use Cases

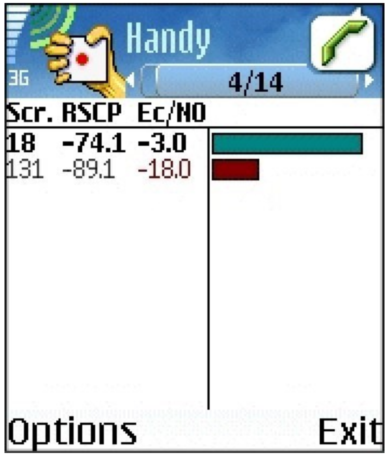
- Network Performance
  - RAN
  - End to end functional
  - Rollout
  - Troubleshooting
  - Optimization
- Network Benchmarking
  - Compare various networks performance
- Device Performance
  - Pre and post launch
  - Lab and field
  - Industrial devices
- Device Benchmarking
  - Compare various devices performance
- Network Monitoring and Assurance
  - Predictive analytics
  - Troubleshooting
  - Roaming

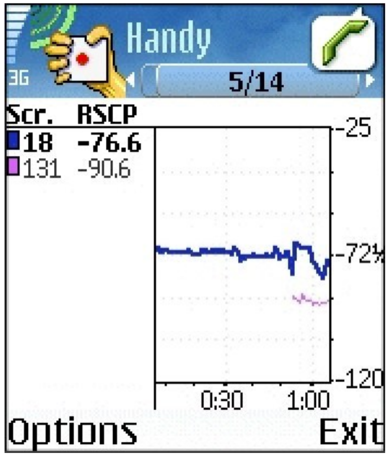
TURKU AMK
TURKU UNIVERSITY OF
APPLIED SCIENCES

Screenshots

Nemo Handy-S Screenshots

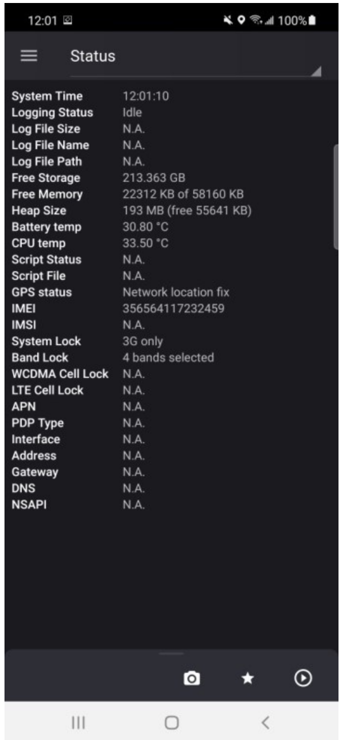


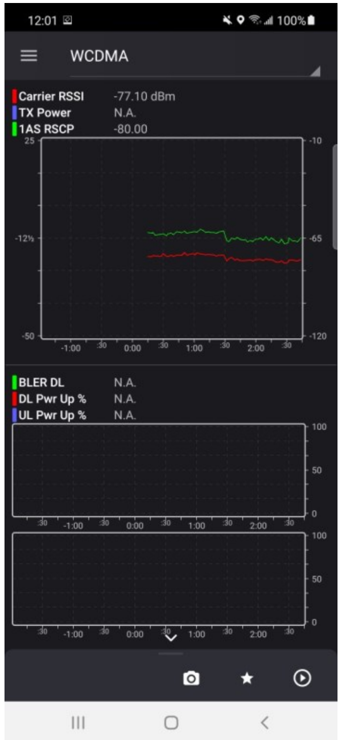


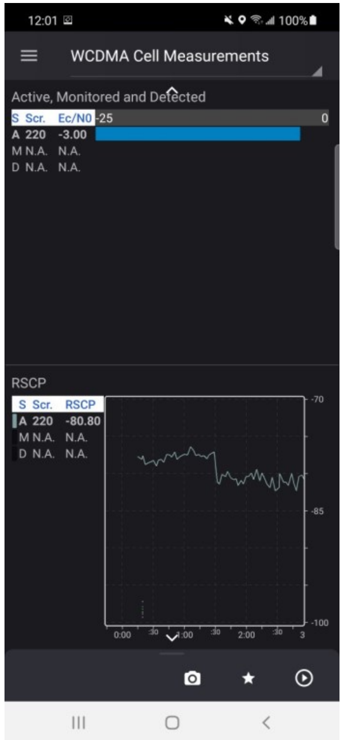


TURKU AMK
TURKU UNIVERSITY OF
APPLIED SCIENCES

Nemo Handy-A Screenshots

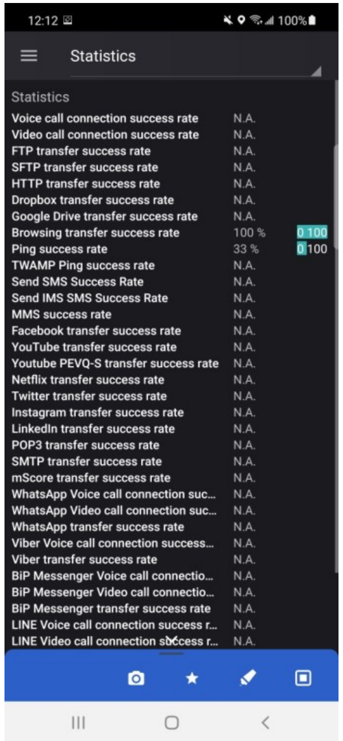


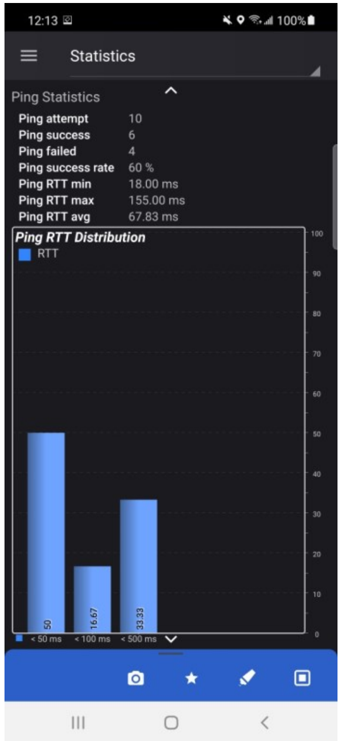


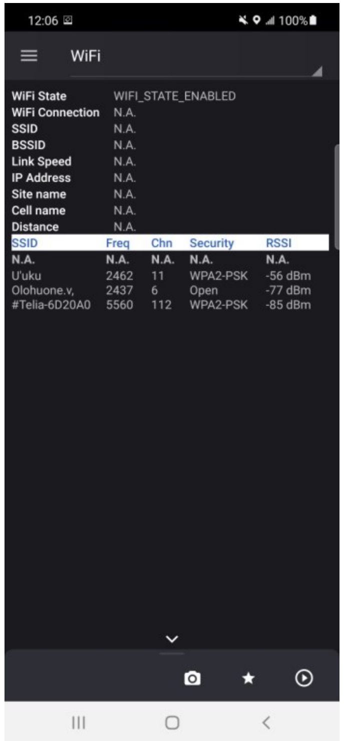


TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

Nemo Handy-A Screenshots

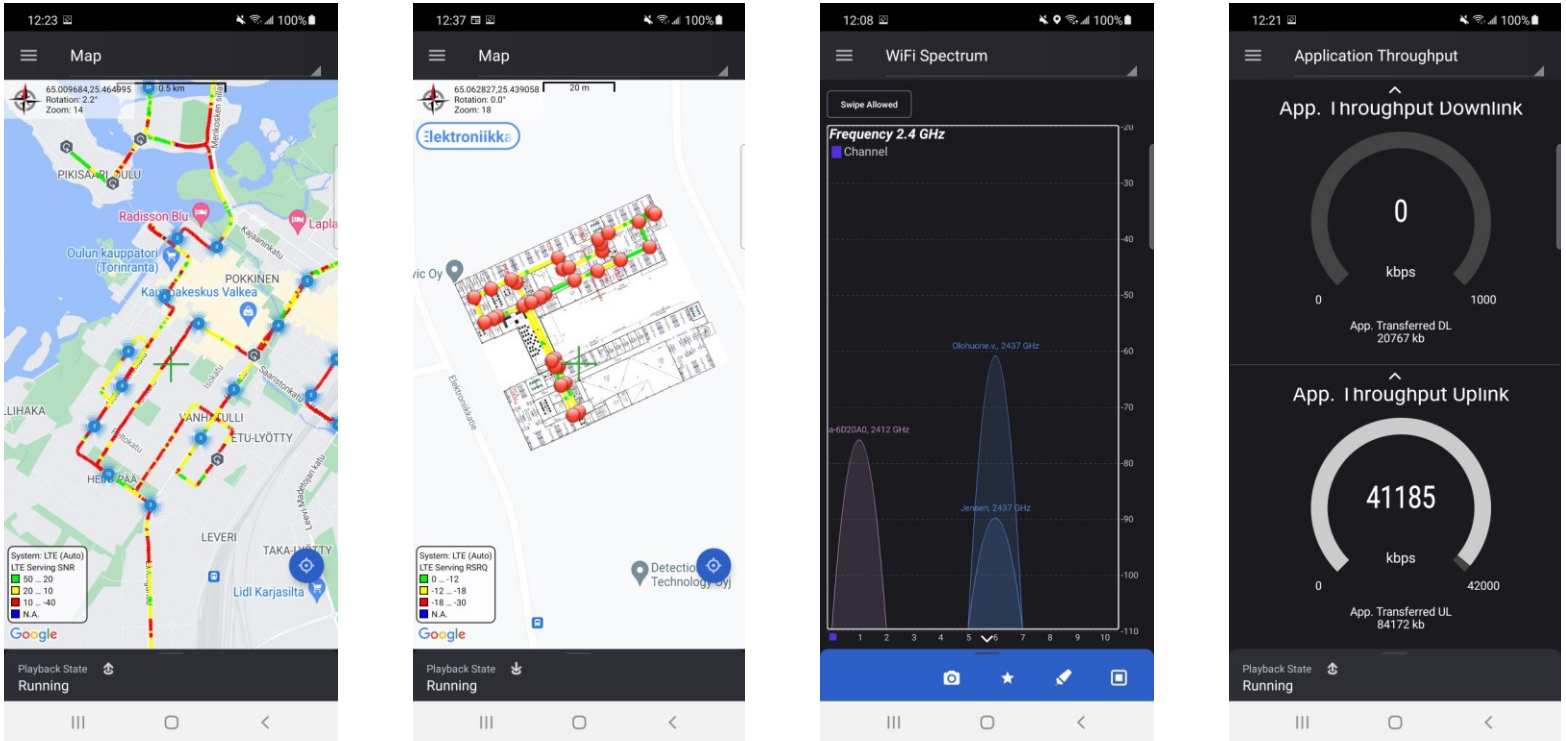






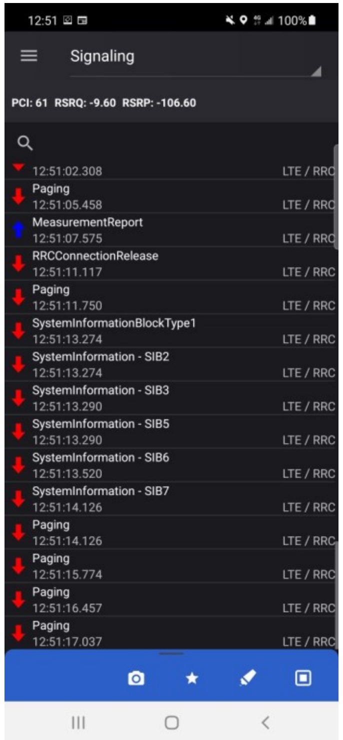
TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

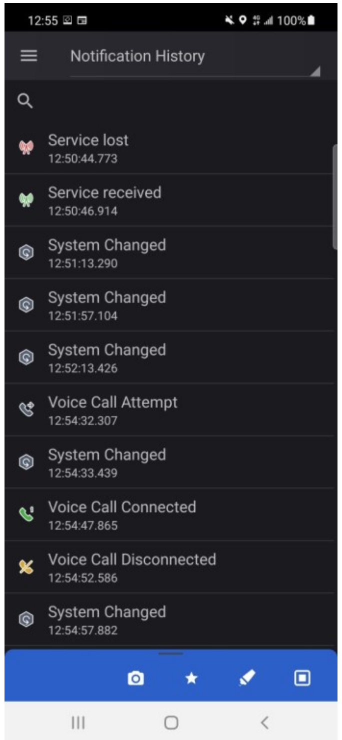
Nemo Handy-A Screenshots

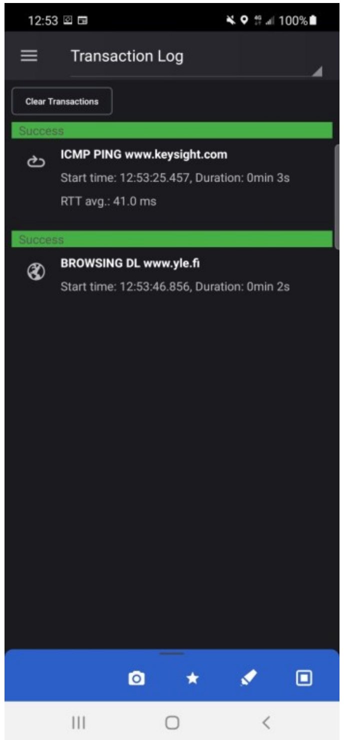


TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

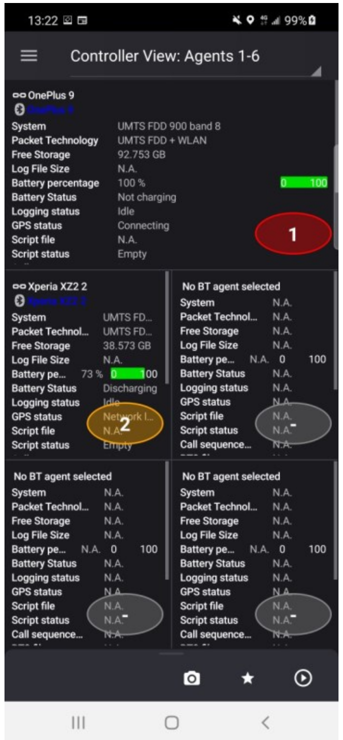
Nemo Handy-A Screenshots

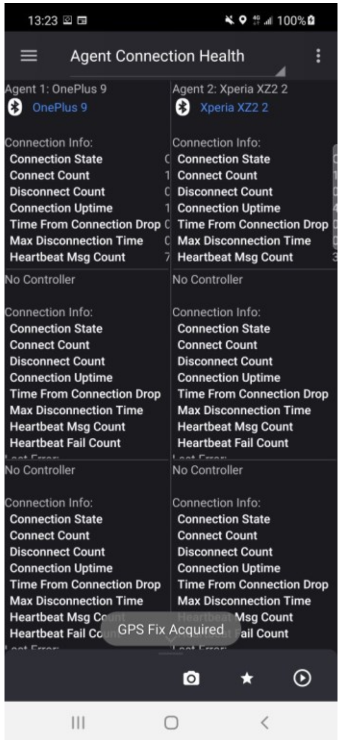






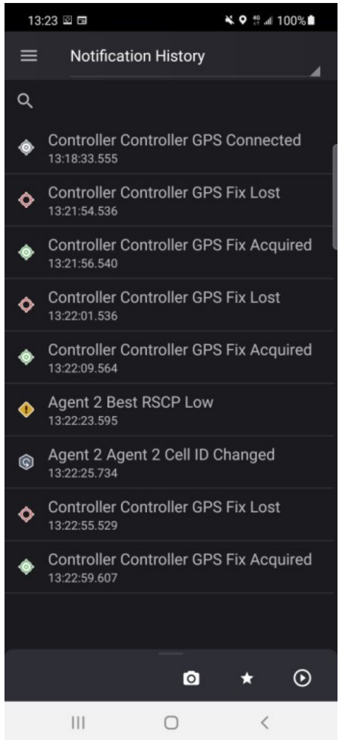
Nemo Walker Air Screenshots

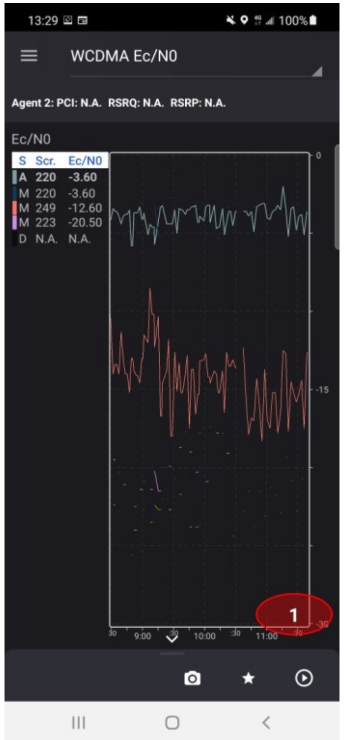




TURKU AMK
TURKU UNIVERSITY OF
APPLIED SCIENCES

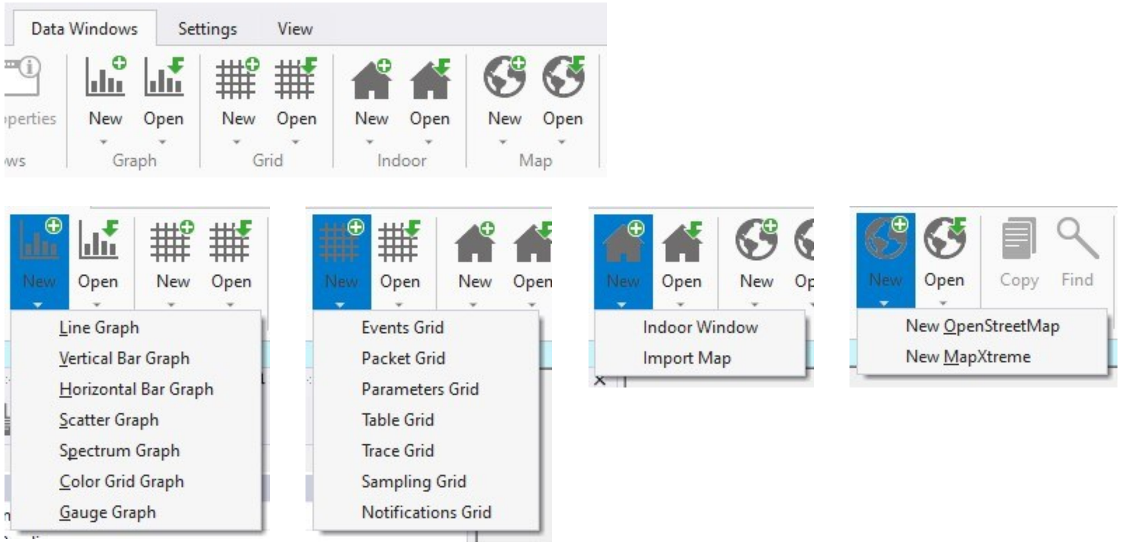
Nemo Walker Air Screenshots





TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

Nemo Outdoor Screenshots

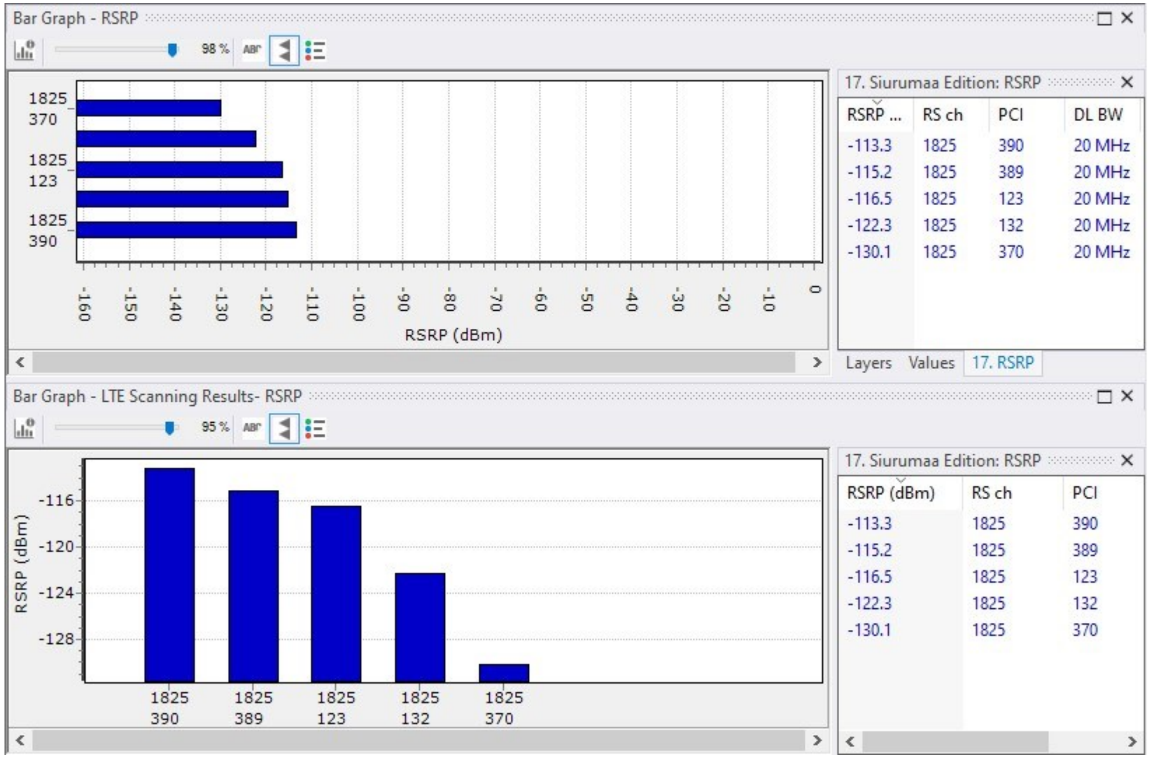


TURKU AMK
TURKU UNIVERSITY OF
APPLIED SCIENCES

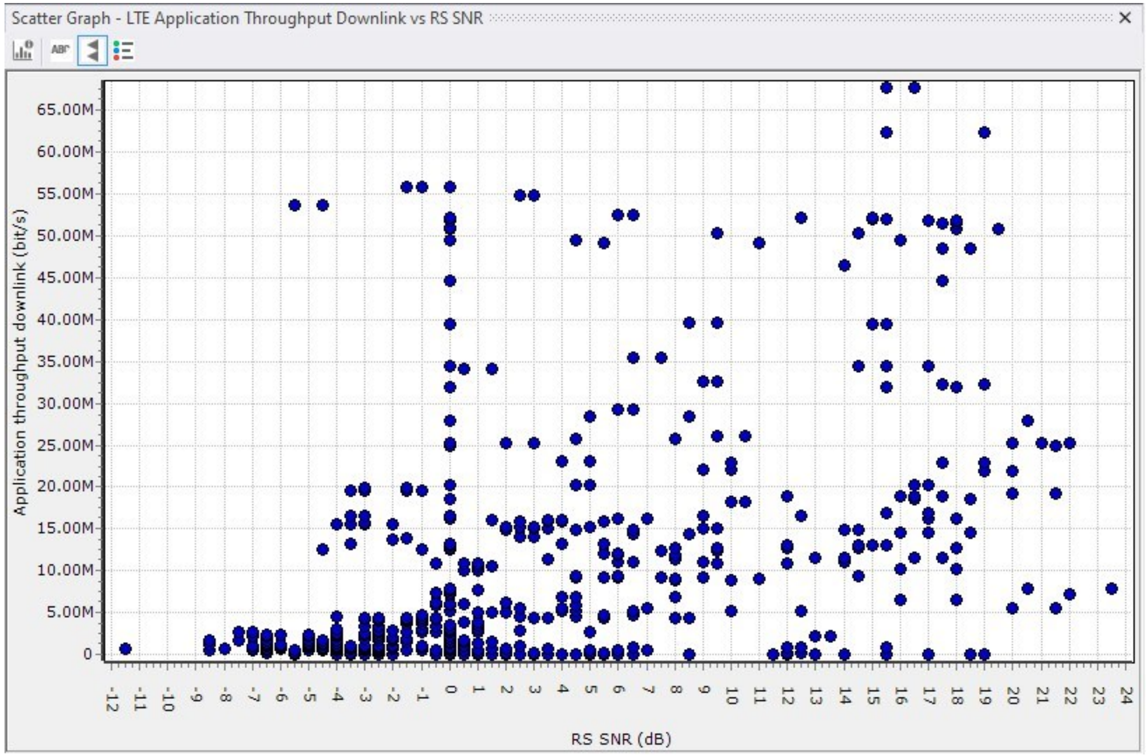
Nemo Outdoor Screenshots

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

Nemo Outdoor Screenshots

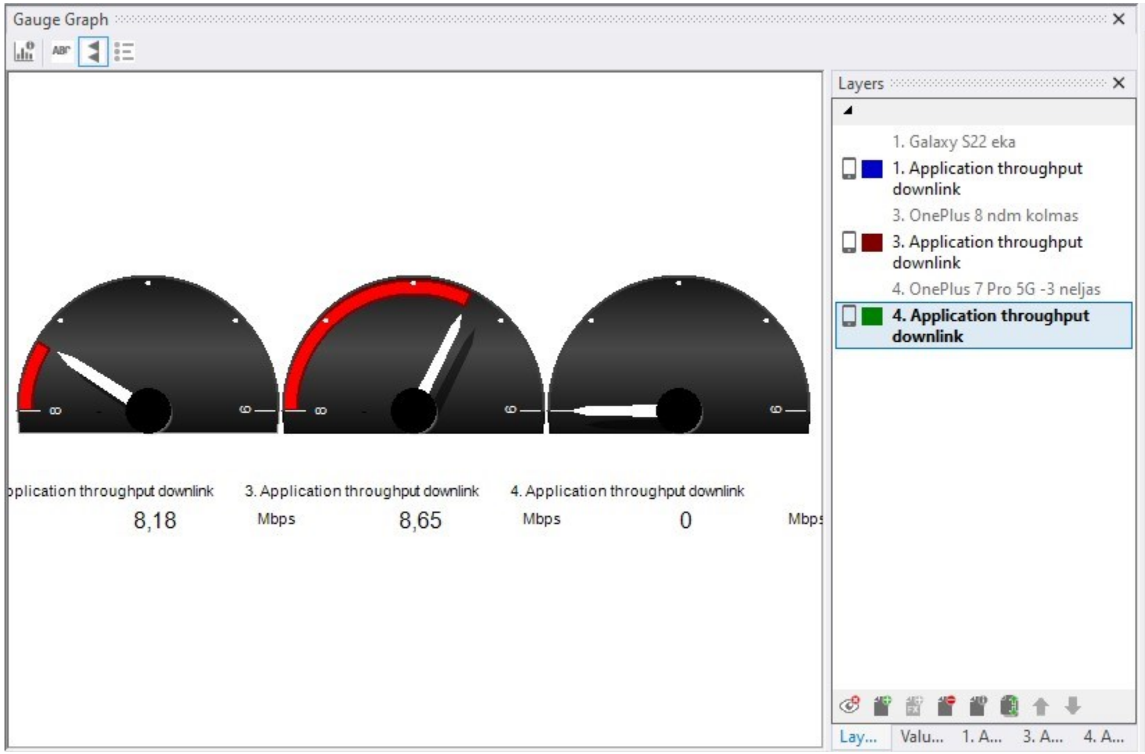


Nemo Outdoor Screenshots



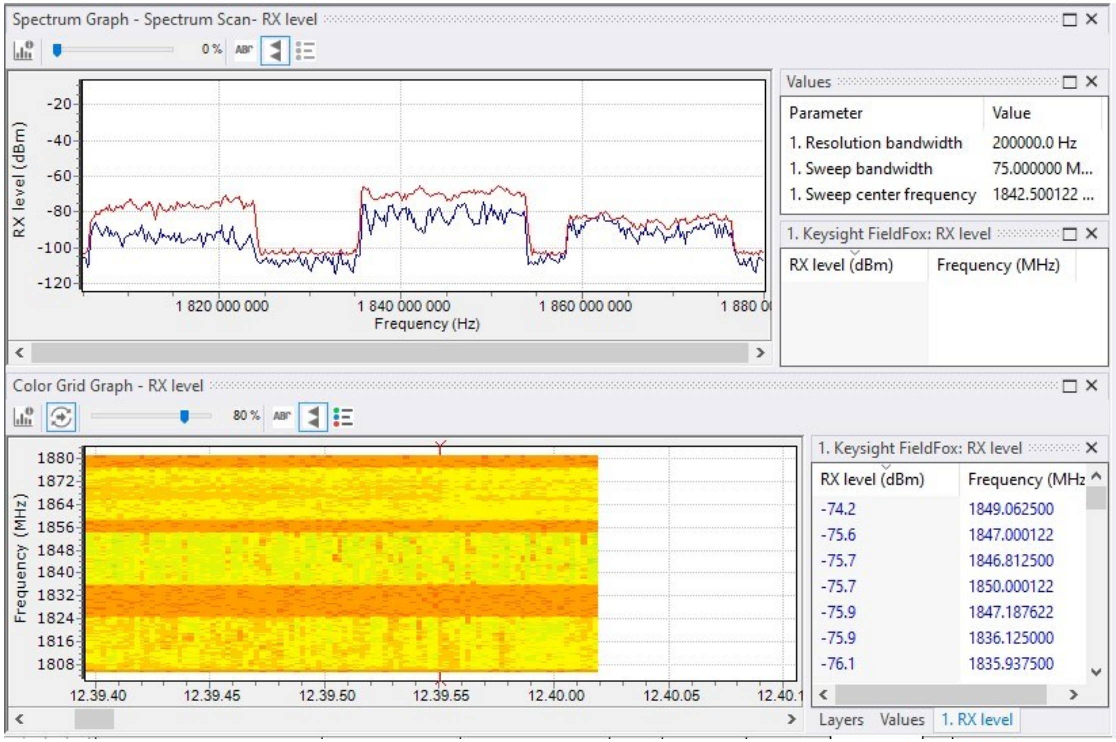
TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

Nemo Outdoor Screenshots



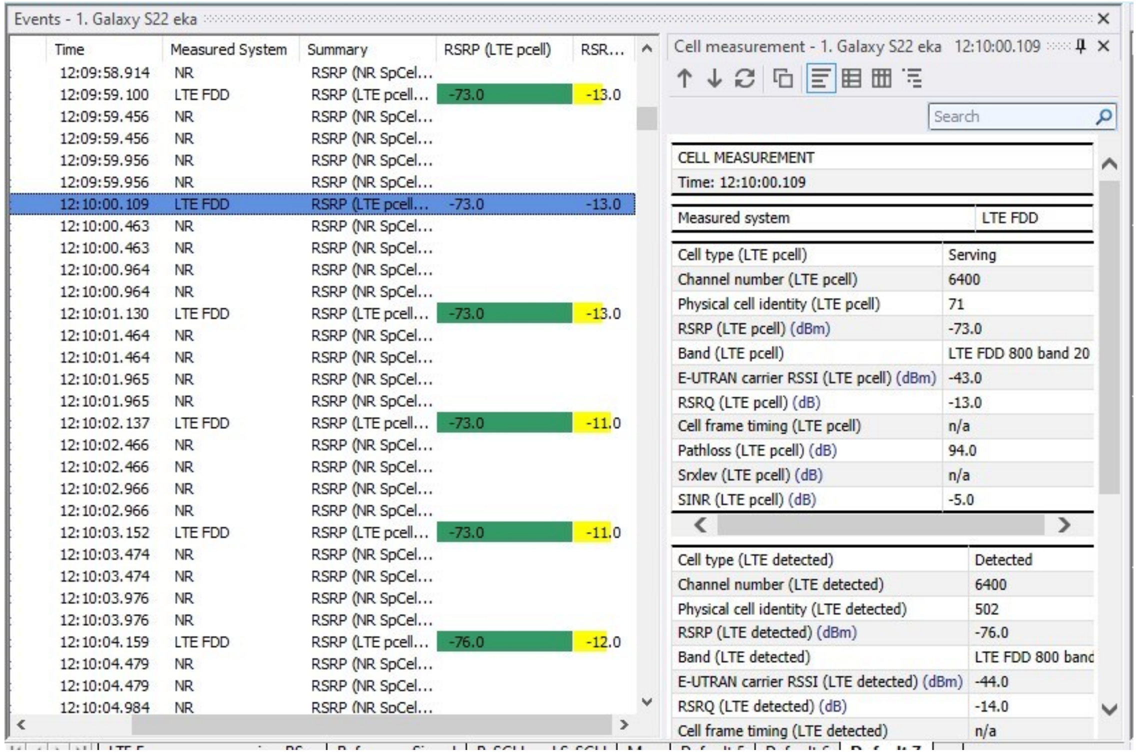
TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

Nemo Outdoor Screenshots



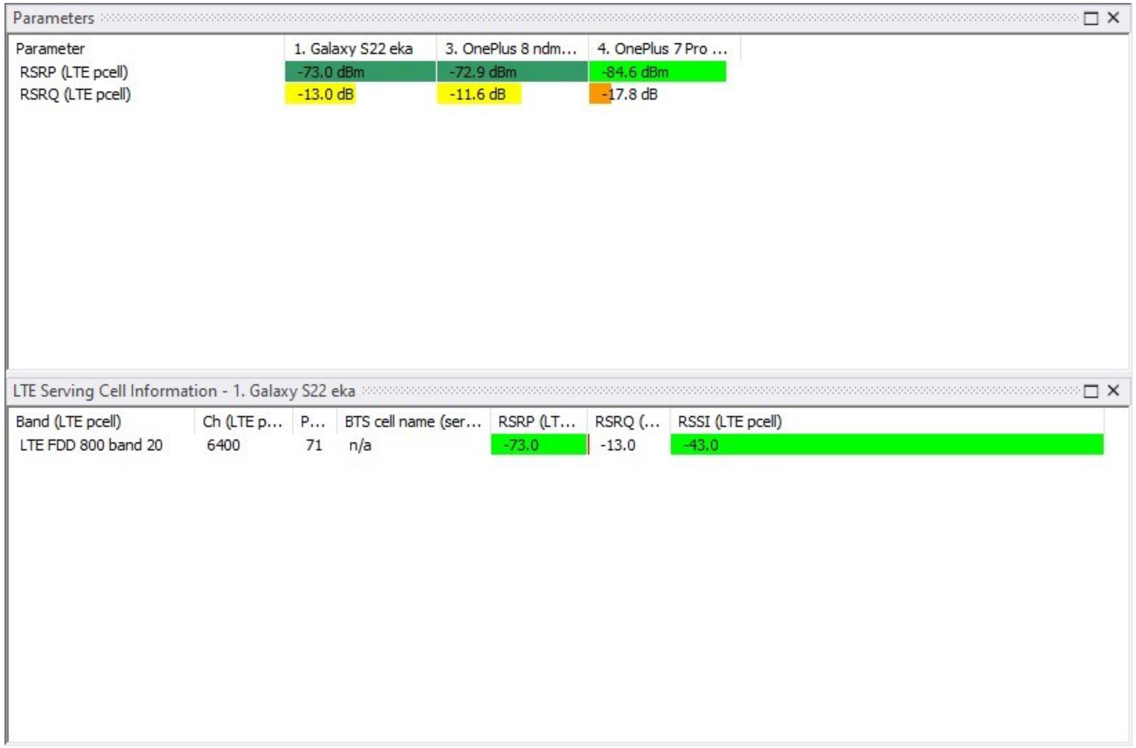
TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

Nemo Outdoor Screenshots



TURKU AMK
TURKU UNIVERSITY OF
APPLIED SCIENCES

Nemo Outdoor Screenshots



Parameters

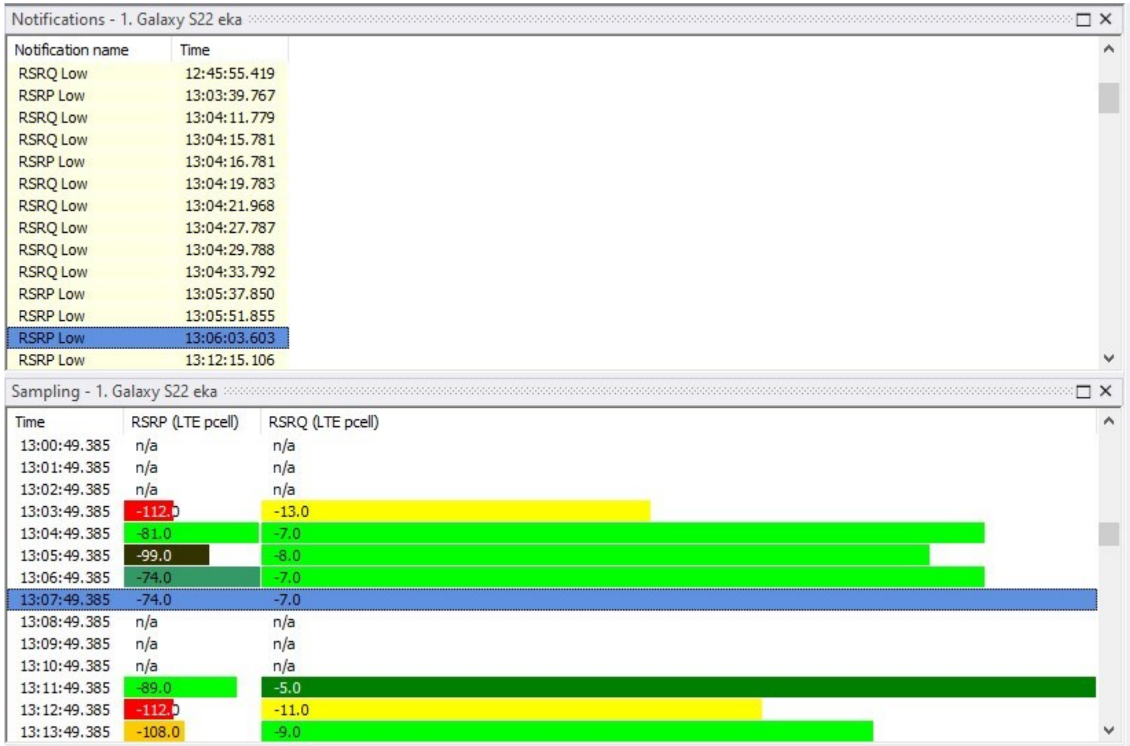
Parameter	1. Galaxy S22 eka	3. OnePlus 8 ndm...	4. OnePlus 7 Pro ...
RSRP (LTE pcell)	-73.0 dBm	-72.9 dBm	-84.6 dBm
RSRQ (LTE pcell)	-13.0 dB	-11.6 dB	-17.8 dB

LTE Serving Cell Information - 1. Galaxy S22 eka

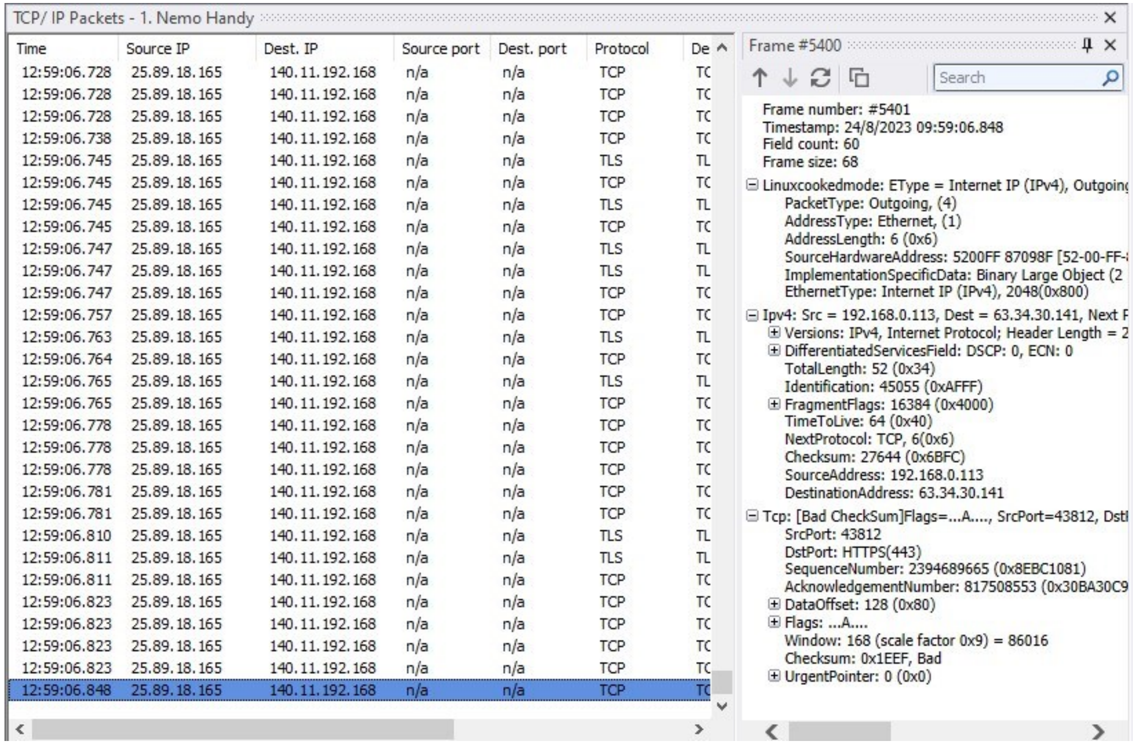
Band (LTE pcell)	Ch (LTE p...	P...	BTS cell name (ser...	RSRP (LT...	RSRQ (...	RSSI (LTE pcell)
LTE FDD 800 band 20	6400	71	n/a	-73.0	-13.0	-43.0

TURKU AMK
TURKU UNIVERSITY OF APPLIED SCIENCES

Nemo Outdoor Screenshots

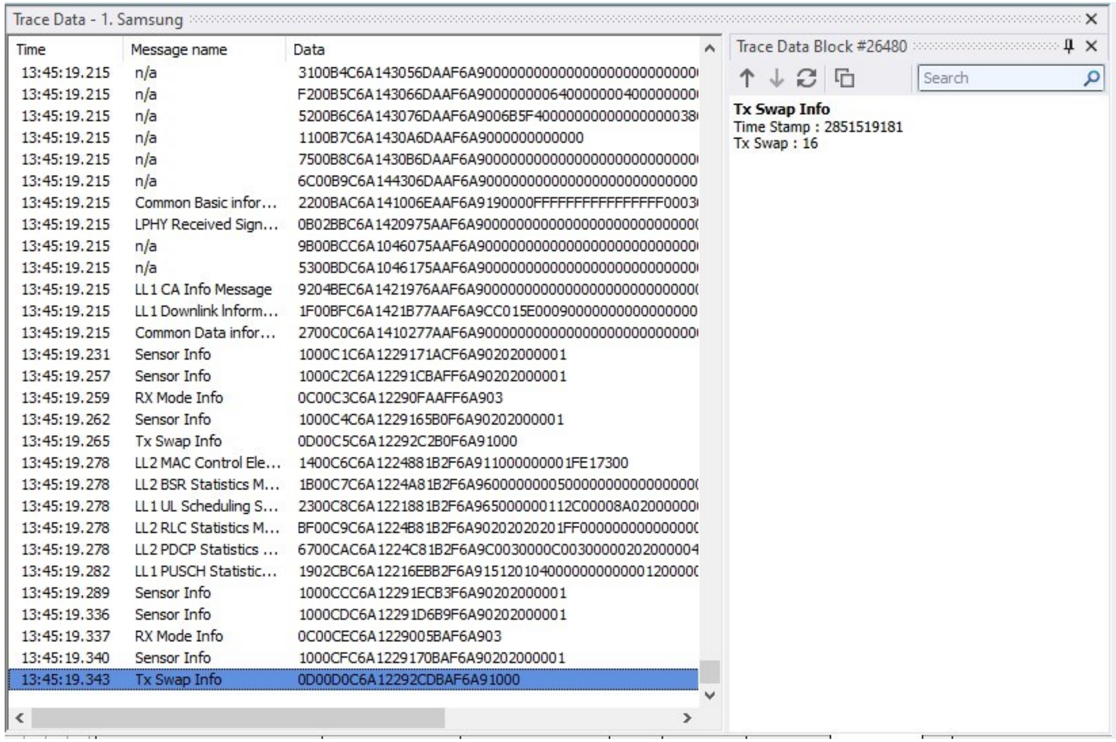


Nemo Outdoor Screenshots

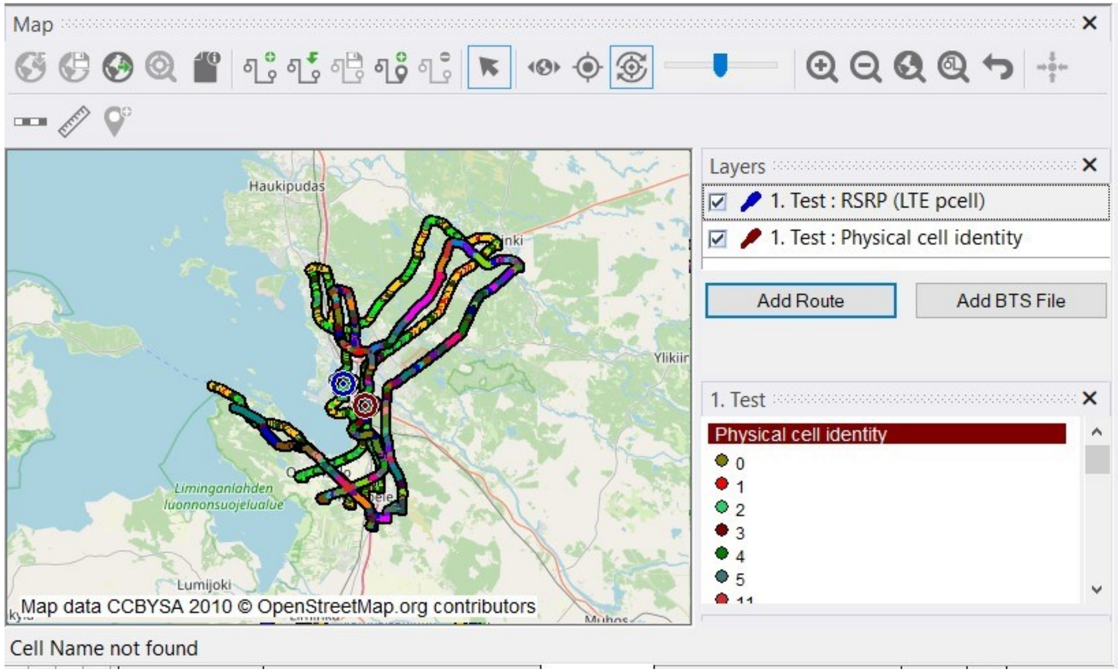


TURKU AMK
TURKU UNIVERSITY OF
APPLIED SCIENCES

Nemo Outdoor Screenshots



Nemo Outdoor Screenshots



Nemo Outdoor Screenshots