Modem Functionality Diagnostic Tool with Graphical User Interface for Cellular Network

Arghya Biswas[1], Sudakar Singh Chauhan[2], Antara Borwankar[3]

[1]School of VLSI & Embedded System Design, NIT Kurukshetra, 136119 India,
[2] ECE Department, NIT Kurukshetra, 136119 India,
[3] Intel Technology India Private Limited, 560103 India,
[1]arghyabiswas05@gmail.com, [2]sudakarnith@gmail.com,
[3]antara.borwankar@intel.com

Abstract: In both Inter of Things and conventional cellular networks the modem is a very important component. So, it should be highly stable and trustable. Any kind of error or misbehavior must be diagnosed and need to resolve properly. In this paper we are proposing a very useful tool that can diagnose the workflow of a cellular modem. The tool can observe the modem very closely and diagnose each step from the basic power on feature to the registration on the network with the establishment of the data connection. If the modem faced any kind of error in between this, then the tool is capable to detect the error and it will solve the problem or recommend the possible solution to get rid of the error. We are also proposing a remote modem diagnosis feature with this tool. If the modem is deployed in some other device on the same network then our tool can also able to diagnose the modem. With this diagnosis tool users can save a lot of time to diagnose and fix the modem error.

Keywords: Cellular modem, Diagnose Tool, AT command, Internet of Things, GSM & LTE network, Yocto

1. Introduction

Cellular Technology is one of the largest technologies in our daily life. This technology was introduced only for voice communication. In the latter days, the same technology was started to use for the data connection. Nowadays cellular technology is being use in various filed. One of the most important filed is the Internet of Things (IoT). The most important part of the cellular networks is the modem (Modulation - Demodulation) [1]. A proper fault diagnostic tool is required to implement, and bug solves the modem related issues. In this paper we are proposing a Functionality Diagnostic Tool of Modem for Cellular Network with Graphical User Interface.

2. Literature Survey

In [2] the authors proposed an On-Board-Diagnostic (OBD) system for Euro Standard Vehicles' Engine. The OBD takes the data related the state of the engine, the exact timing of when a spark happens etc and send the data to a remote server. They used the GSM technology with the AT command to communicate with the server. A Graphical User Interface (GUI) on the server-side shows the coming data and suggests some required action if any error occurs.

The authors proposed in [3] a multi-node mesh network to analyze the data from DTE. They grab the data from one node and send it to the other nodes and then the system dials up to initiate the cellular connection. In other node the modem must send the ASCII or string response according to the command that fired. They chose WSN topology for communication and some shell to troubleshoot the issues.

In [4] authors used to send a 2.5GHz signal after 15 minutes gap over a corn and soybean field for 7 months. After analyzing the RSSI, Rainfall, Temperature etc they concluded that the RSSI is affected by the preset and absent of the crops.

Authors proposed to share the lower layer ID IMSI for several IoT devices in [5] to optimize the state information. It will be recognized by the upper layer devices. They also proposed to use time division concept in Machine Type Communication to upload the data in different time to enhance the speed of the system.

Authors proposed in [6], Machine to Machine services to implement smart resemble metering in the application for the vehicle. They also measure the quality of the signal overhead and analyze the data. With the proposed system the mainly focused on reducing the signaling load on the cellular network sever.

In [7] authors proposed a tool to collect the signal information and control plane message to analyze the performance degradation due to the mis-configuration of cellular network. In these analyzing steps like time threshold, control flow sequence, and signaling failure, authors used more than 3.1 million control plane messages from the 13 main cellular operators to analyze the circuit-switched fallback technology in 3G and LTE system.

In this paper [8] authors summarized the main issue in the 5G technology evolution for fault management. To overcome those issues the authors proposed an application from an operational perspective based on Machine Learning. They describe the details workflow of the application. They use the deep learning process to survey the issues and based on that they also build the block diagram for the typical fault management system.

According to the authors [9], their proposed system can able to diagnose the fault in the cellular network for the IoT devices. They proposed a data-driven indoor diagnostic framework for this application. This framework uses the machine learning technique and trained some prediction models. Those models can able to construct a global view of the radio environment which is well known by its name Radio Environment Map (REM). Later this REM will be used for diagnosing and managing the faults in the cellular network at the location.

In [10] authors proposed to combine the Self-Organizing Networks (SON) and traces. According to them, it will help to detect the issued related to the radio interface and easy to solve the issues. As the call traces functions have the user information so it will be easier to detect and resolve the issue automatically with this proposed model. This model also helps to assess the RF condition based on the location data.

3. Motivation

All the projects that are based on the cellular network have a common component, called. This modem is an essential part of GSM comm. In the IoT applications, the modem keeps also a big contribution. So, we should concentrate on the functionality of the modem. And we should make sure that the modem should work perfectly under all circumstances. But if it faces any error then we should also be prepared to fix the issues. So, for this point of view, we are motivated to make a proper functionality diagnostic tool with a graphical user interface for modem cum cellular networks.

4. Research Gaps

Conventionally a cellular modem is diagnosed by manual method. The test modem needs to be connected with a computer via a carrier board. Then AT commands are serially sent to the modem from the computer one by one. The person responsible for the diagnose observer the response from the modem and decide the reason of error. So this procedure is too much time consuming, and sometime the person need to remember all AT command and the command sending sequence. So, we feel a need for an automated tool with a graphical user interface that can send all those AT commands and grab the response from the modem and sometime the tool can try to solve the error by its own.

Another big gap that we found is remote access to the modem. If the modem is deployed as part of a product in a remote location, then it is a bit difficult to access and diagnose the modem if any issue occurs. So, in this project, we are also trying to focus this remote access part also.

5. System Overview

5.1. System Architecture

The modem functionality diagnostic tool is designed for the UNIX based operating system only. The GUI part will be installed in the local Linux system. But for the hardware, there are two possible options. If the modem is in the local location, then it can be connected via a carrier board through the USB port. But if the modem is deployed in the remote location then one client and one host script must execute to connect and communicate over the server.

In our setup, we were using the Intel XMM7560 modem device. One USB type carrier board was used to communicate with this modem. If the modem is in the remote location them the system must connect over the server to communicate with the modem. For this setup, we used the Intel IA board. And the modem is connected via the PCIe interface with the Intel IA board. The board itself connected with the server. In our case, we test it over the LAN server. For the WAN we need some network side changes.

5.2. System installation

The entire modem diagnosis tool package contains the diagnostic tool binary itself and the dependency libraries & an auto-installation script. The package has different binaries for the different mode of modem. For the remote mode the target must be set up before install the tool. Firstly, the UNIX system and the Intel IA board must be connected on the same local server and able to ping each other. In the installation process, the installer will ask for the mode selection. According to the mode it will install the correct binary, the dependency scripts and libraries. Figure 1 shows the mode selection process.

```
arghyabi@All-Series:$ ./mmgui_installer.sh
Starting Installer Script...
Checking Internet Connectivity..
Internet Connectvity available


SELECT THE USE MODE OF MODEM-MAMAGER-GUI
 1 => Modem-Manager-gui with ofono
 2 => Modem-Manager-gui with atmanager
 3 => Modem-Manager-gui with WHL atmanager
```

Fig.1. Installation mode selection

For the remote or PCIe mode, we must go through some extra steps after installation. After successful ping in both host and platform, a couple of binary need to run in both the system like fig 2 to initiate the socket [11] connection.

```
 <-- create_socket -->
 Socket already created
 Thread creation sucessfull
Thread creation sucess = 0
Connect Failed -1
 <-- sock_write_task -->
 blocking on accept
 <-- create_socket -->
socket initialised 6
Connect sucessfull 0
 Blocking on read
```

```
 Thread init sucessfull
 Thread set state  sucessfull
 <-- at_read_task -->
 Blocking on read at channel
 Thread creation sucessfull
Thread creation err = 0
 <-- sock_read_task -->

Blocking on accept
 accept port number 48322
accept successed socket ID = 5
 Blocking on read socket
```

Fig.2. Log for Host socket and Remote socket binary

The next step is common for the both mode of operation. In this step, the tool binary is being executed and opens the GUI. For the PCIe mode the tool checks some extra things like device's presence, device drivers, network interface, AT ports, and modem state before send the first AT command. If the tool found any problem here, then it will report the error.

Now the tool will send AT command, grab the response for the command and show in the GUI. First, few commands will grab the information about the modem, SIM, PIN, network, etc. Then it registers the SIM with the live network and read the network type, signal strength, operator name, SPN, context-related data, and location info. Fig 3 is the AT command flow.

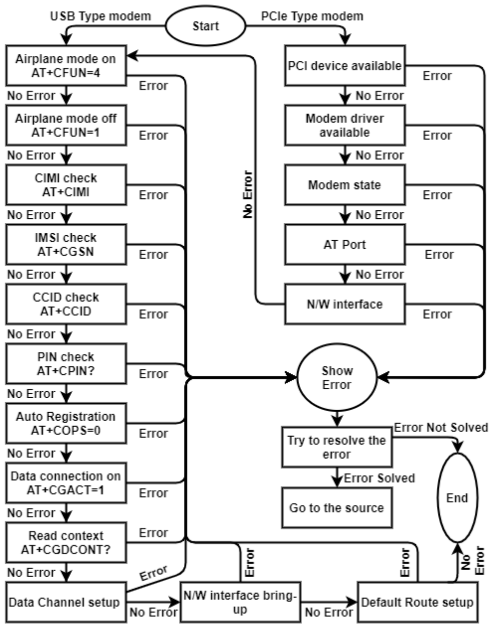

Fig.3. Diagram for the full system

If all the response is okay and doesn't have any issue then the tool down the network interface for the modem and put the IP address, subnet mask and default gateway on that network interface and finally up the same. Users can verify the network interface by ping any global IP.

After every step, the tool grabs the response. If any error came in the response, then it sends the error response to the error checking block and tries to resolve the error. If the error is resolved, then the tool came back to the next AT command else, it will suggest the possible step to resolve the error.

6. Implementation Details

The hardware implementation is different for both USB and PCIe mode. In the USB mode, a carrier board is needed. So, for this configuration UNIX is PC able to communicate with the modem directly over the AT ports which automatically come up when the board is connected to the PC.

For the PCIe mode, the Intel IA board must be connected with the UNIX PC via LAN with the successful socket communication. As the modem is connected in the PCIe bus so the PCIe drivers are also needed.

Fig 4 is a screenshot of the tool, contains basic information like device port, IMEI, IMSI, roaming status, operator name, registration status, registration mode and network signal strength 3GPP type location, MCC, MNC, LAC, TAC and CID. Some elements may change the value WRT time and location. So, these values are continuously updated in the GUI.

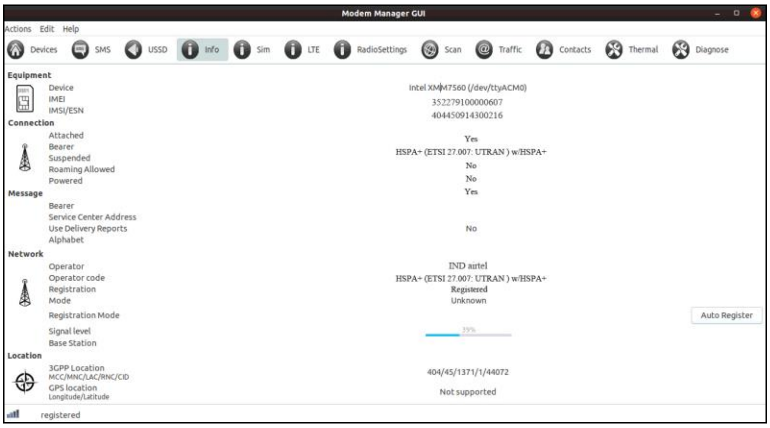

Fig.4. Info Tab of the Tool

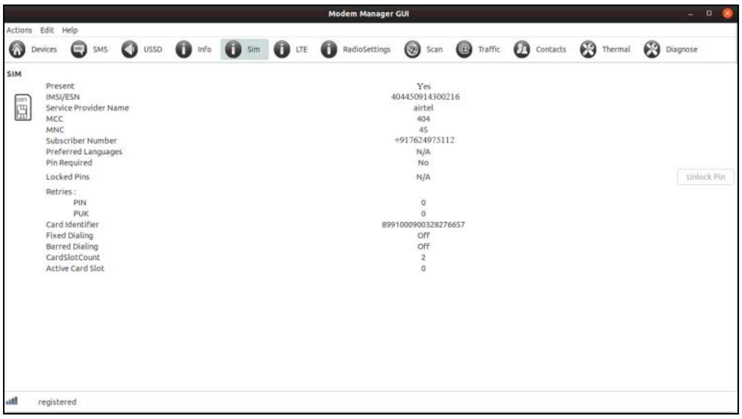

Fig.5. SIM Tab of the tool

Fig 5 is the SIM tab that contains the SIM details like SIM presence, IMSI, service provider name, MCC, MNC, subscriber number, PIN required or not, PIN entry retries SIM card identifier, card slot count and the current card slot number. This tab also has the features to unlock the PIN.

The diagnose tab in figure 6 contains the PCIe device name, drivers needed for the modem, network interfaces, AT ports, modem status. This tab also contains the context for the modem and two buttons to refresh the data and resolve the current issue. Figure 7 is also showing the diagnose tab with some error messages. Somehow the modem has been timed out and the AT ports are also not available in it. In this case, the issue may solve by clicking the resolve button.

Most of the cases the tool automatically resolves the problem by clicking the resolve button. But if it is beyond the scope of the tool then it suggests the steps and hopes for the user interfere.

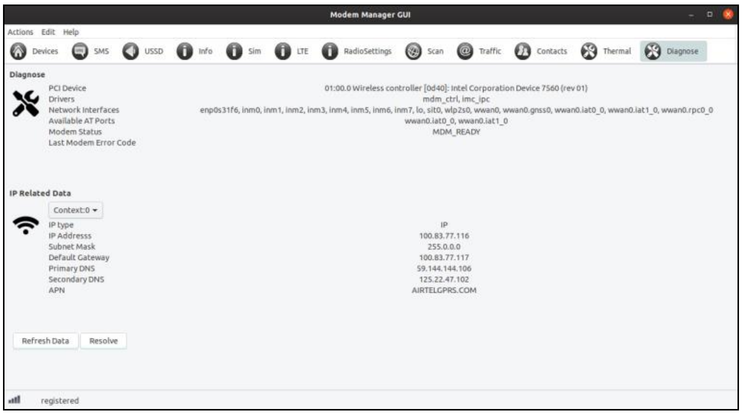

Fig.6. Diagnose Tab of the tool

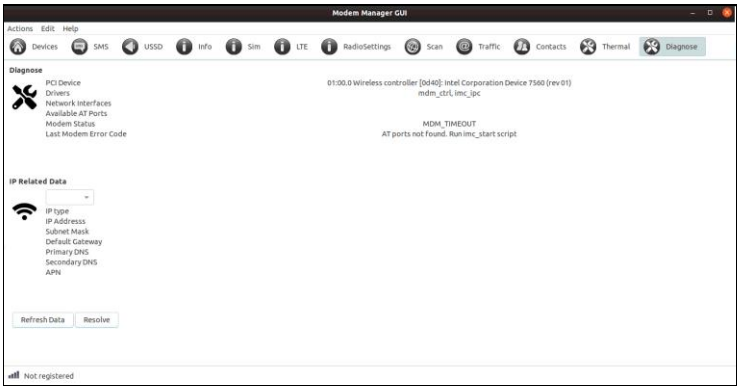

Fig.7. Diagnose tab for the tool with error message

7. Conclusion & Future Work Scope

Modem functionality diagnostic tools can be used to diagnose the modem related issue in different types of hardware platforms. This will reduce the time to manually diagnose the modem by sending AT command one by one. This tool is designed in such a way that it can work with other plugins also. So, this plugin feature gives the versatility of the software. Till the date this tool support three main plugins like Modem-manager, Ofono, and AT-manager. The diagnose feature is available only in the AT-manager plugin.

This tool is totally free and open source. So, anyone can download and modify it. This tool still has some scope of up-gradation or future work. For the PCIe type modem, we need an extra Linux based computer. The tool needs to run in the host Linux PC. User need to create a network socket connection in the both Linux PC and the Intel IA system. If the socket connection is successfully built then the tool can communicate with the modem through the socket connection over the IP address. But in the future, it is also possible to embed the tool itself on the Yocto [12] recipe with the dependencies and build the OS for the Intel IA platform. Then the extra socket connection is not further needed.

In this diagnose tool some modem manager feature is discarded, and only focus on the data session diagnose. So, it is also possible to add the modem manager features like Messages, Contacts, Radio Frequency, etc. And it is also possible to add the diagnose feature for these features also. Then it will be a full feature modem diagnosis tool.

Reference

1. Kwon H, Kim K, Lee C (2011) The unified UE baseband modem hardware platform architecture for 3GPP specifications. Journal of Communications and Networks 13:70–76. doi: 10.1109/jcn.2011.6157254
2. Iqbal MN, Xin LY, Rehman WU, et al (2017) Diagnostic tool and remote online diagnostic system for Euro standard vehicles. 2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC). doi: 10.1109/itoec.2017.8122328
3. Alyazidi A, Frej MBH, Gupta N (2018) Multi-node partial-mesh network communication via AT commands. 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT). doi: 10.1109/lisat.2018.8378013
4. Hunt KP, Niemeier JJ, Cunha LKD, Kruger A (2011) Using Cellular Network Signal Strength to Monitor Vegetation Characteristics. IEEE Geoscience and Remote Sensing Letters 8:346–349. doi: 10.1109/lgrs.2010.2073677
5. Ito M, Nishinaga N, Kitatsuji Y, Murata M (2016) Reducing State Information by Sharing IMSI for Cellular IoT Devices. IEEE Internet of Things Journal 3:1297–1309. doi: 10.1109/jiot.2016.2587823
6. Kouzayha N, Jaber M, Dawy Z (2017) Measurement-Based Signaling Management Strategies for Cellular IoT. IEEE Internet of Things Journal 4:1434–1444. doi: 10.1109/jiot.2017.2736528
7. Hong B, Park S, Kim H, et al (2018) Peeking Over the Cellular Walled Gardens - A Method for Closed Network Diagnosis -. IEEE Transactions on Mobile Computing 17:2366–2380. doi: 10.1109/tmc.2018.2804913
8. Basangar S, Tripathi B (2020) Literature Review on Fault Detection of Equipment using Machine Learning Techniques. 2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM). doi: 10.1109/iccakm46823.2020.9051543
9. Chou S-F, Yen H-W, Pang A-C (2019) A REM-Enabled Diagnostic Framework in Cellular-Based IoT Networks. IEEE Internet of Things Journal 6:5273–5284. doi: 10.1109/jiot.2019.2900093
10. Gomez-Andrades A, Barco R, Munoz P, Serrano I (2017) Data Analytics for Diagnosing the RF Condition in Self-Organizing Networks. IEEE Transactions on Mobile Computing 16:1587–1600. doi: 10.1109/tmc.2016.2601919
11. Xue M, Zhu C (2009) The Socket Programming and Software Design for Communication Based on Client/Server. 2009 Pacific-Asia Conference on Circuits, Communications and Systems. doi: 10.1109/paccs.2009.89
12. Navik AP, Muthuswamy D (2017) Dual band WLAN gateway solutions in Yocto Linux for IoT platforms. 2017 International Conference on Internet of Things for the Global Community (IoTGC). doi: 10.1109/iotgc.2017.8008968