CLIENT WIFI ASSURANCE TOOL

APPLICATION NOTE

This application note is designed for a wireless assurance tool that runs on an enterprise end point, monitors the SOPHOS WIFI connectivity experience. The data to measure experience and troubleshoot the system is read from the WIFI driver inbuilt in the system. The first step is that the application need to be able to scan for access points within the range of the device and capture the results of that scan. Next step is that the application has to scan for the connected network. The connected network as well as radio scanning information is saved in the database and the information will show up on the browser(localhost) and based on this information, gauge graph is populated showing user experience and troubleshooting table is displayed.

**Key Terms**

(W)AP – (Wireless) Access Point: A device that allows the user to connect wirelessly to a network.

BSSID – Basic Service Set IDenitifier: This is another name for a MAC address.

dBm or dBmW – deciBel milli-Watts: a signal strength rating in decibels as a power rating compared to 1 milliWatt

MAC address – Media Access Address: The physical address of a network device.

SSID – Service Set IDentifier: A name given to a network to describe it.

Wi-Fi – Wireless Fidelity: The protocol most frequently used for wide local area networks.

RSSI - Received Signal Strength Indication: This is a dBm value that indicates the wireless signal strength received by the client and it usually ranges from 0 to -100. The higher the value, the stronger the signal, being 0 the weakest and 100 the strongest

SNR - Signal-To-Noise Ratio: This is measured in dB that is used to compare the signal received with the background noise. The higher the value, the better the communication quality.

Channel - Network operating channel. If the network is operating over more than one channel, all operating channels are displayed here.

DHCP – Dynamic Hosting Configuration Protocol: This is a network management protocol used to dynamically allocate an Internet address to any device or any node on a network.

DNS – Dynamic Naming System: An Internet service that translates domain names into IP addresses.

Gateway – A gateway is a hardware device that acts as a “gate” between two networks.

Latency - It is the time required to transmit the packet from host to destination in a network. It can be measured in one way or roundtrip.

Jitter – It defines the variation in the delay of received packets.

Throughput – It defines the quantity of data being sent or received per unit time.

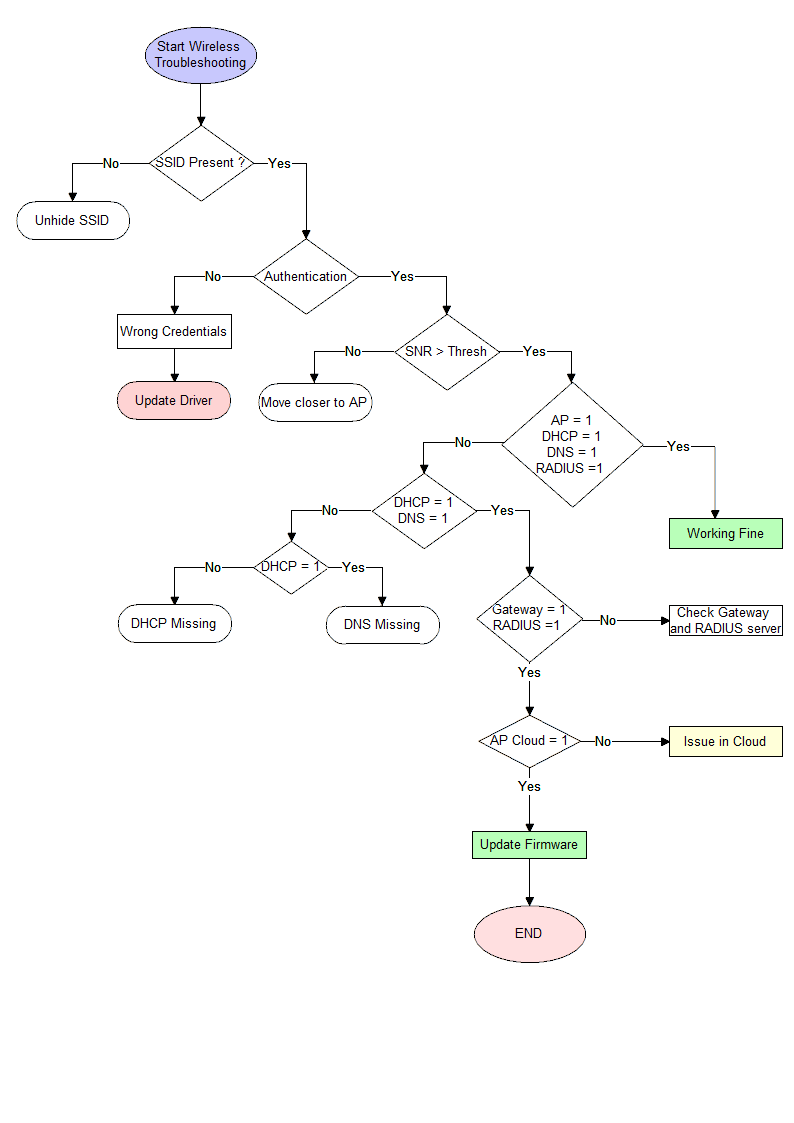
**Introduction**

Wi-Fi is a critical resource of businesses of all types and sizes. It is no longer the exclusive domain of larger enterprise or home networks. Whether it is a retail shop or a five-person startup the reliability of the Wi-Fi connectivity is an important connectivity factor. At the same time, the challenges in delivering consistent Wi-Fi performance and availability continue to grow. This include technical factors such as increased usage driven congestion as well as organizational factors like lack of expertise or resources in the business to manage Wi-Fi.

**Objectives**

The purpose of the application is tocheck and verify WIFI parameters related to layer1 and layer 2 which include basic ping test for above layers entities (RADIUS server, DNS server, DHCP server and Gateway) to ensure proper working of WIFI. Based on radio scanning and ping test, gauge graph and troubleshooting table would populate indicating user experience and Wi-Fi parameters respectively.

**Error-Flow Diagram**



Explanation

1. Staring with wireless troubleshooting, the first step would be to see for the SSID – if it is present or hidden. If SSID is hidden, it should be made unhidden to let users connect to it. If SSID is present and still client is facing issues connecting to it, it could be due to some issues with NIC card of user device. Rebooting device should solve the problem.

Sometimes issues could be on AP side that it got hang up or goes into continuous reboot cycle due to low PoE (Power over Ethernet) which could be solved by rebooting AP or ensuring proper PoE.

1. Next step would be to see for the proper authentication credentials. It could be SSID name or password misspell or it could be that device authentication and encryption standard does not match with that of AP. Still if the client faces issues with Wi-Fi, client’s device drivers need to be updated.
2. Next, we can move on to check for wireless parameters like SNR - If that is below the set threshold, the client should consider moving closer towards AP. Other parameters like Transmission rates, receiving rates, PoE, Channel Utilization and non-interfering channels should also need to be taken care of (Refer correlation logics for further insights).
3. Moving on, the ping connectivity to DHCP server, DNS server, Radius server and Gateway should be ensured for the proper working of Wi-Fi services. Further AP cloud connectivity can also be taken care of to ensure smooth functioning of whole Wi-Fi environment.
4. If you encounter no issues in radio parameters as well as underlying infrastructure, may be AP firmware might be the issue. Updating it would serve the purpose.

Also in highly dense environment, if there is high use of AP resources i.e. CPU and memory utilization, client might experience denial of services.

1. Apart from design and infrastructure issues, parameters like throughput, latency, jitter, DFS frequency, airtime, retries and roaming could play their part in foiling your attempts to successfully run your Wi-Fi services (Refer correlation logics for further insights).

**Background**

Tools required to build the application

Front-end

HTML: It provides the basic structure of sites, which is enhanced and modified by other technologies like CSS and JavaScript.

CSS: It is used to control presentation, formatting, and layout.

JavaScript: It is used to control the behavior of different elements. Here we have included condition for co-relation logics.

Bootstrap: It is an open source toolkit for developing with HTML, CSS, and JS. Quickly prototype your ideas or build your entire app with our Sass variables and mixins, responsive grid system, extensive prebuilt components, and powerful plugins built on jQuery.

jQuery: It is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

Plotly: It creates leading open source tools for composing, editing, and sharing interactive data visualization via the Web. We are using plotly library to visualize the User Experience Meter.

AJAX: It is a developer's dream, because you can update a web page without reloading the page. Request data from a server - after the page has loaded. Receive data from a server - after the page has loaded. Send data to a server - in the background

ANGULARJS: HTML is great for declaring static documents, but it falters when we try to use it for declaring dynamic views in web-applications. AngularJS is a toolset for building the framework most suited to your application development. It is fully extensible and works well with other libraries. Every feature can be modified or replaced to suit your unique development workflow.

Server

Node: It is providing open source environment and used in cross platform. It uses JavaScript on the server. Here is how Node handles a file request: Sends the task to the computer's file system. Ready to handle the next request. When the file system has opened and read the file, the server returns the content to the client.

Database

Sqlite3 is a popular choice as embedded lightweight database software for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems (such as mobile phones), among others. Sqlite3 has bindings to many programming languages. Sqlite3 used for storing data of Radio Scan, Current Scan, and AP Scan modules.

Back-end

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Python supports modules and packages, which encourages program modularity and code reuse. Python 3.6 is used to develop cross-platform back-end script and to build Machine Learning model.

**Correlation Logic**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter | Optimal Range | Condition | Experience | Technical Inference | Suggestions |
| RSSI | -65 for Video  -70 for Data | RSSI < -75 | <= 4 | Low signal strength. | Move closer to your access point. |
| -65 to -75 | 5 to 7 | Medium Strength |  |
| -50 to -65 | 8 to 10 | Good strength |  |
| PoE | 22– 25 mW  Very High Density – 1mW | >= 33 mW | < = 4 | High Interference, Sticky Clients, Roaming Problems, Hidden Nodes | Reduce Power Range |
| 20 to 33 mW | 5 to 7 | Medium Interference |  |
| Below 20 | 8 to 10 | Nominal Interference |  |
| Rx | 600 –700Mbps | < 400 | < = 4 | Low receive rate | Move to other AP or upgrade your device’s technical specifications . |
| 400 to 600 | 5 to 7 | Optimal receive rate |  |
| > 600 | 8 to 10 | Good Receive rate |  |
| Tx | 350-450Mbps | < 300 | < = 4 | Low transmit rate | Possibly Legacy System. Upgrade your device or increase device power. |
| 300 to 500 | 5 to 7 | Optimal transmit rate |  |
| > 500 | 8 to 10 | Good transmit rate |  |
| Frequency | 5 GhZ | DFS (5.25 to 5.33 Ghz) | < =4 | Denial of Service (Radar Use)  Normal Usage (Otherwise) | Keep checking for DFS alert and migrate to other channels on positive alert. |
|  |  | 2.4 Ghz | 5 to 7 | Legacy Clients | Upgrade technological specifications of your device. |
|  |  | 2.4/5 Ghz | 8 to 10 | Desired | Good Connectivity |
| Bandwidth  [Channel Utilization] | 20% of total available bandwidth | >= 80% | < =4 | Impacts whole Wi-Fi communication | Check streaming activity (if any) or move to another AP |
| 50 to 80% | 5 to 7 | Impacts video communication |  |
| 20 to 50% or less | 8 to 10 | Impacts voice communication |  |
| Authentication | AES | TKIP | <=5 | Legacy Clients | Upgrade to higher encryption standard |
| AES | 6+ | Newer Standard |  |
| DNS | Reachable (Ping test successful) | 0 | 0 | No translation of IP address to domain names | Verify DNS connectivity |
| 1 | 10 |  |  |
| DHCP | Reachable (Ping test successful) | 0 | 0 | Device would be active only on LAN n/w but can’t communicate outside it (APIPA) | Verify DHCP connectivity |
| 1 | 10 |  |  |
| Gateway | Reachable (Ping test successful) | 0 | 0 | Unable to access the internet | Verify Gateway connectivity |
| 1 | 10 |  |  |
| Radius | Reachable (Ping test successful) | 0 | 0 | No access to network resources | Verify Radius Server connectivity |
| 1 | 10 |  |  |
| Latency | <30ms | >100ms | < =4 | Degrades performance noticeably | Reduce Latency by decreasing retransmissions |
| 30 – 50ms or less | 5 to 7 | Video Applications |  |
| <30ms | 8 to 10 | Voice Applications |  |
| Jitter | <5ms | <5ms |  | For VoWiFi |  |
| ~ 100ms |  | For data |  |
| Throughput | 70% of WiFi speed | 70% | < =4 | For streaming data | Reduce user density. Also see for network overheads. |
| 50 -70% | 5 to 7 | Data applications |  |
| <50% | 8 to 10 |  |  |
| Airtime |  | As low as possible |  |  | Check data rates |
| Retries | 5% | >10 % | < =4 | Unacceptable retransmission for all communication | Check Latency, Jitter and reduced throughput |
| 5 – 10% | 5 to 7 | Good for transmitting data |  |
| < 5% | 8 to 10 | Good for VoWiFi |  |
| Roaming  [Hands Off] | 50 ms | As fast as possible |  |  | Check if FT is enabled and set standard to ‘Voice Enterprise’ |

**Deployment**

The technologies involved in the project consists of a database equipped with frontend support which can scan the radio signals around the system and store the scanned data properly in the database.   
First, we need to start the node server by running the node script which specifies the port information. The script also has a python script running in background scanning the radio signals & current scanning and storing in the specified SQLite database. This DB is created automatically by the python script.   
 Next, we make calls to retrieve the parameters stored in the SQLite database and give it to frontend as JSON object. JSON are portable and easy medium of transferring data through web. This JSON is handled by the frontend code which is totally written in Angular JS.

Now to deploy the project, make sure you have downloaded all the files with the folder hierarchy preserved & maintained and perform following steps in order:

**Installations on Windows:**

* Install Python3.x from <https://www.python.org/downloads/>**.** Add it to your path as environment variable.
* Install any code editor or preferably Visual Studio Code.  
  Visit <https://code.visualstudio.com/>.
* Download SQLite from “https://www.sqlite.org/download.html” and follow the instructions given at <https://www.tutorialspoint.com/sqlite/sqlite_installation.htm> for installation.

Type **sqlite3** in command prompt to check if it’s working.

* Install Node JS from <https://nodejs.org/en/download/>. Add this also to your path as environment variable.
* Install node package manager and initialize required packages by typing “**npm init**” in the windows command prompt shell. This generates a package.json file with necessary dependencies to be installed.
* For executing Python scripts in node JS file, we need node package manager “**npm**” to install python-shell package in node. Open windows command prompt and type “**npm i python-shell**”.
* In the same prompt execute “**npm install sqlite3 express**” to install corresponding dependencies.

After all the dependencies have been installed, open the directory where the code files are present.

* Move the **index.html** and **logic.js** files into public folder.
* Run the Node JS Server file which starts a server in specified port.

Open a command prompt and type

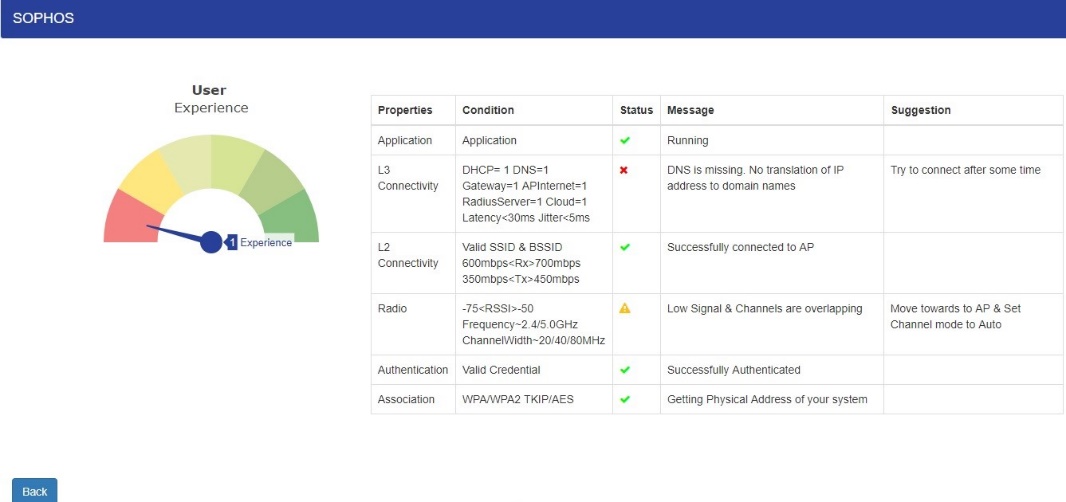
“***node server.js*** “

* This server JS file has a python script which can scan the radio signals and store in SQLite3 database.
* The python script has terminal command “**netsh wlan show net mode=bssid**” for radio scanning and “**netsh wlan show interface**” for current scanning which are executed in the background.
* Now open the **index.html** file on the specified port as,

“ <http://localhost:8080/index.html>”

* Now the webpage loads up. Press the Scan button to view the results.

**Screenshot**



**References**

*Ekahau Special Edition: Wi-Fi Network Design for dummies by A Wiley Brand*

*Wlan Troubleshooting by David.D.Coleman,CWNE #4*