ARUBA UTILITIES

ArubaUtilities is an Android application designed as a testbed for techniques relating to AP selection and handover, site coverage surveys and determining location. It is designed to run on Android Rev 2.2 and later.

Aruba Utilities includes a number of tools useful for characterizing and troubleshooting wireless LANs from Aruba Networks. Some tools work with any WLAN, others depend on Aruba’s Airwave management system. Aruba Utilities includes:

* A Handover Monitor showing current access point, dynamic signal strength and RSSI measurements, other access points audible to the device and handover events
* A Floorplan view that downloads the floorplan picture and AP details from your network’s Airwave WLAN management system. See where APs are located relative to your position, and touch AP icons for details of current loading, channels and power.
* The Floorplan view also offers a locally-generated estimated heatmap and a site survey function that links actual coverage measurements to indicated positions on the floorplan.
* An implementation of the Airwave Management Client reports device information and scanned APs to your Airwave WLAN management system.
* An integrated version of the Iperf application allows for easy throughput testing.
* Measurements are written to a plain-text log file that can be saved for use later.
* Various testbed functions compare different ways of estimating location from Wi-Fi and GPS.

Aruba Utilities was developed by the CTO Group in Aruba Networks as a testbed for our research into WLAN measurement and optimization techniques. It will be of interest to network engineers with Aruba Networks WLANs, particularly those with Airwave.

Each of the five tabs in the application is largely self-contained, we will treat each in turn:

HANDOVER TAB

This is a tool to show how the device sees its Wi-Fi environment, optimized for a multi-AP enterprise WLAN. It takes major inputs from two sources. First, the current Wi-Fi connection is monitored every 2 seconds, and its signal strength (RSSI in dbm), as reported by the Wi-Fi driver, is shown on the screen in green. Note that the RSSI figure is generally reported by the Wi-Fi chip without knowledge or calibration of the RF amps and antenna chains in the receive path, so two different devices side-by-side are quite likely to give different results – the absolute calibration should not be trusted to better than ~4dB in our experience. The grey histogram display walks leftwards every 2 seconds, so 20 seconds of RSSI history is shown. Underneath the RSSI figures the current data rate of the connection (modulation rate) is shown in green, in Mbps.

At the top right of the tab we show the currently associated BSSID, and the duration of this association. When the device hands over to a new AP the histogram display has a grey box underneath it to indicate the handover event, and the information for the current, last and previous associations at the top of the tab are walked leftwards. This top row thus shows the recent history of associations.

Underneath the histogram section we show the last scan results. This is a different source of data from the current connection. Every 10+ seconds, or on-demand, the Wi-Fi chip is put in scan mode, where it goes off-channel and sends probe requests, waiting for responses from neighbouring APs before moving to the next channel. In ArubaUtilities we kick off a scan event every N seconds, driven by the menu item **Force Scan Int (default 10sec)**. it’s generally not useful to set this shorter than 10 seconds, as the Wi-Fi chip can be busy and won’t respond on-demand. The main ArubaUtilities menu (not the preference or settings menu) has a **Force Scan** button that can be used to kick the scan function into life. Even that doesn’t always get a response from the Wi-Fi chip if it’s busy trying to find an AP to handover to.

The scan event returns a list of BSSIDs, SSIDs, channels and signal strengths (RSSI, dBm). This list can be filtered to only those BSSIDs advertising the currently-associated SSID (menu item **Filter Scan Results**): it’s sometimes useful to shorten the list, for instance when testing near the Aruba QA lab where 80+ BSSIDs are audible in the 2.4GHz band. In addition to the parameters above, the BSSIDs are marked with an A if their OUI resolves to Aruba Networks (via an internal lookup table) and a \* for the currently-associated AP. If there is a successful connection to AirWave (see the FLOORPLAN tab), and if the AP is named in AirWave, the name will be shown.

Since the current-connection parameters and last-scan results are from different reports, they often differ; this is to be expected, and is an indication of signal strength fluctuation on the air and measurement errors in the chip. Also, where there are many APs or large amounts of Wi-Fi traffic, the scan will often miss APs even when their signal is strong. This indicates the difficulty of getting fast, accurate information about the Wi-Fi environment from a client device on the WLAN.

FLOORPLAN TAB

The floorplan tab shows the WLAN topology, using information obtained from the AirWave AMP and VisualRF services. This tab uses various techniques to locate the device in both local (x, y) and global (Lat, Long) coordinates, and there are other functions such as a locally-generated heatmap. We use this as a testbed for evaluating client-side location techniques, and these functions will be improved, and more added, over time.

AirWave Interface

The AirWave server offers a comprehensive API through which much of the information seen on the VisualRF screens can be obtained parametrically. For this tab to function, the AirWave server needs to be accessible over Wi-Fi and the user must have an account on the server. Also, the device running ArubaUtilities should be associated with an AP on an AirWave floorplan (see below for one exception to this rule).

Since Rls 7.0, AirWave uses a persistent-cookie http login method over an SSL connection. ArubaUtilities implements SSL (with a ‘trust all certificates’ function, so beware) with an Apache HttpComponents 4.1 <http://hc.apache.org/index.html> http client. There are menu entries for the AirWave hostname **Domain** (e.g. <https://your.airwave.com>), **User Name** (myusername) and **Password** (mypassword). This should be all that is required to reach AirWave. The login and cookie function is probably the most error-prone function in ArubaUtilities, not because it’s particularly buggy but because so many things can go wrong. For instance, if the AirWave server’s clock or timezone is set wrong, the cookie may be expired on-delivery, and login will never be successful. We are adding diagnostics to this function as we come across these events; it’s always a good thing to check that PC- or Android-based browser login works in the event of failure of ArubaUtilities. For now, we display a ‘toast’ message if a valid cookie is not obtained.

Once AirWave login is successful, ArubaUtilities downloads information about the client’s location, the APs and the floorplan bitmap using AirWave’s XML over http API. The sequence is broadly as follows:

1. Look up the device’s MAC address in VisualRF to see if it has a valid location from the VisualRF location engine. This will only be true if enough time has elapsed (~10 – 20 minutes) since the device first authenticated on that floor of the building (‘site’ in AirWave terms). If a valid location is found, it is downloaded along with the reference to the current site ID – usually each floor of a building is a site.
2. Look up the device’s MAC address for the associated APs MAC address (BSSID) in AMP (strictly, the AMP and VisualRF functions are separate within AirWave). This generally converges much faster than the VisualRF location. The result is the site ID where the associated AP resides. Note that if the device associates to an AP on a different floor, this site ID will be incorrect while the VisualRF location above will almost always correctly resolve the floor. To tie-break such a disagreement, there’s a menu item to trust AMP (**Floorplan from my AP**) where the alternative is to trust VisualRF.
3. With the site ID as key, we look up the site and obtain information on all the APs. This includes their type, location, current transmit power, and how each receives the signals of other APs – AirWave keeps extensive AP data. There are also references to the building ID and floorplan ID, used below. Note that the locations of APs are set by drag-and-drop in AirWave. If APs are mis-placed over the floorplan in AirWave, our subsequent calculations will be off.
4. A lookup to the building ID is used to find the Lat, Long coordinates and orientation (north\_heading) of the floorplan. This information can be entered in AirWave from Rls 7.5. On the floorplan view, use the ‘Building Orientation’ function to set Lat, Long for two points on the floorplan and VisualRF will translate to the Lat, Long of the orgin (and client locations) and the north\_heading. To see the XML report, press ctrl-shift-x-m-l when in building view.

We will use this Lat, Long and Heading data to translate from local (x, y in feet from the top-left vertex of the floorplan diagram) to global (Lat, Long like GPS) coordinate systems. One quick way to get started is to use Google Earth or another mapping function to find Lat, Long for features of a building in satellite view: it is usually possible to identify from the floorplan where its corners would be in an aerial view of the building. If the Lat, Long is not set in AirWave (or not giving correct bearing, which has been my experience up to August 2012) use the ‘**Map Origin Latitude**’, ‘**Map Origin Longitude**’ and ‘**Compass Offset**’ entries in the ‘Settings’ menu to enter them for your floorplan (Lat and Long are in decimal degrees for the top-left point of the floorplan as seen in Airwave; offset angle is in degrees between true north and the positive y-axis of the floorplan).

1. The floorplan ID is used to download a bitmap of the floorplan. This is the bitmap uploaded to and used by AirWave. We provide menu options for **Thumbnail Floorplan** or full bitmap. Generally, we find thumbnail provides sufficient resolution for small screens, while Android sometimes runs out of memory when trying to download large or detailed full-size bitmaps in the hundreds of kB range. So we recommend staying with the thumbnail option initially.

All this data can take some time to download. The http GETs work in an Async Task, which puts them in the background and doesn’t suspend the UI, but it can easily take a minute for initial download under bad conditions, such as a weak or noisy Wi-Fi signal (or testing next to the Aruba QA lab). The process repeats till it is successful, but once the data is downloaded successfully we won’t go back to get it again, so if ArubaUtilities comes up with the wrong floorplan, or you move to a different floor of the building, exit and start it again. Let us know if this is a problem – it can be fixed, but the logic is quite complicated. However, since VisualRF regularly recalculates client locations, we repeat the sequence above every (menu item **Location Update Int** seconds).

The result of a successful download will be a scaled floorplan on the screen, showing AP locations as small blue circles with names. This should reflect the floorplan view on AirWave.

Once the floorplan is downloaded, we can locate the client using different techniques:

1. From VisualRF (menu item **Airwave**). This is a network-side calculation. In brief, it takes AP-AP calibration measurements and constructs a heatmap, then applies the vector of AP-reports of each device’s signals received at that AP to find the most likely location. There are many nuances in the algorithm, it is very sophisticated but RSSI triangulation is the basic mechanism. This method is generally very accurate, because the WLAN is better at hearing the client’s transmissions than the client is at stimulating and receiving probe responses from APs, but because the polled-SNMP link between AirWave and WLAN controllers is relatively infrequent, updates are usually at 5 – 10 minute intervals.
2. Client-side measurement s(menu item **AP Signal Strength**). As an indication of what the client sees, we colour the associated AP in red and APs that the last scan identified in solid blue. We display circles based on the signal strengths reported from the last scan event. The scarcity and volatility of these circles gives an idea of how difficult it is to use client-based RSSI measurements for location, and why most algorithms heavily average scans to prevent location jitter – averaging is at the expense of fast updates for moving clients, of course.
3. While simple signal strength measurements are the basis for all Wi-Fi location today, we can do better than the method above, which uses a simple estimate of AP transmit power (~20dBm) and a linear signal strength vs distance relationship. The state-of-the-art is to use Gaussian Processes, mathematical techniques that fit curves to multi-dimensional data, to build a signal strength heatmap. ArubaUtilities implements this model, using as inputs the transmit signal strength of each AP (reported from AirWave, as in modern WLANs the transmit power will be set by automatic algorithms such as Aruba’s ARM and will often be less than the regulatory maximum) and the reported measurements taken by other APs of its beacons. These figures are used as input to the Gaussian Process along with a simple signal propagation model: the goal is to use as few special algorithmic twists as possible. All calculations are done on the device in a background Async Task that runs after a successful download of AP and floorplan data from AirWave. The resulting heatmap can be displayed through a menu item (**Display Heatmap**).
4. Once the heatmap is obtained, a local indication of location is calculated, based on the vector of APs heard in the last scan (as seen on the HANDOVER tab). The menu option to display this is **GP Model**. Every 40 seconds or so (menu item **Location Update Int**) the calculation is repeated and the solid green square is placed at the most likely location. Because the list of reported APs and their signal strengths varies from scan to scan, this location bounces around a lot – we haven’t optimised it yet. One attempt, taking the maximum signal strength readings from APS in the last N (usually 5) scans, shows some promise but is still not reliable. We will improve this algorithm.
5. Android devices have 3 methods of determining location. If GPS is enabled and a satellite fix is obtained, a Lat, Long location is usually accurate to within 10 metres. In the absence of GPS, Android uses celltower triangulation and can look up a central database for a match against scanned Wi-Fi BSSIDs. Even indoors, this mix of location functions usually gives a reasonable result. For test purposes, we display it as a blue solid square on the floorplan, translating the Lat, Long to x, y coordinates. A menu item (**GPS**) switches this on.
6. Android has a useful function called ‘Mock Location’. This allows an app like ArubaUtilities to inject a location into the Android location engine described above, by masquerading as the GPS provider. This can be useful because mapping applications like Google Maps use the location API to find where they are, and even when GPS does not have a fix, the Mock Location function can feed a good location. ArubaUtilities has a menu item (**Set Mock GPS Location**) that takes the locally-derived location (the solid green square) and refreshes mock location every 10 seconds. When this is active, Google Maps will display the ArubaUtilities location rather than that from the GPS receiver. (The Android menu function ‘Allow Mock Location’ must be set for this to work.)
7. Android devices include a compass. This can be used to help orient the user, if a menu item (**Rotating Floorplan**) is selected. The floorplan should be rotated appropriately, based on the north\_bearing explained above. If no north\_bearing figure is available, the menu item **Compass Offset** can set the floorplan’s orientation.

Getting an accurate, fast location, especially for moving devices, is one of the most difficult problems in Wi-Fi and we will be extending the functions above to evaluate include techology such as device-based inertial navigation and round-trip-time distance measuements in the future. If the goal can be met, site surveys will no longer require any user intervention to identify waypoints, which most systems must use today.

The floorplan tab attacks another of the significant challenges of enterprise WLANs – visualizing the network. The heatmap deals with one aspect – whether there’s adequate coverage for each spot on the floor – but troubleshooters will want more. The basic floorplan tab overlays AP locations and names (obtained from Airwave). An optional setting (**Show AP Details**) shows the channel and tx power for each radio (usually two radios per AP) next to each AP icon. The setting is provided because the screen can get cluttered. With this option it’s possible to see all channel and power assignments simultaneously, to get a feel for the RF plan. There’s a further option. If an AP is touched, the heatmap of only that AP is shown, and we also display a number of current parameters of that AP. All of these are obtained from Airwave, which in turn retrieves them from APs/controllers every few minutes using SNMP.

1. Bandwidth and client count per-AP (across both radios if two in use)
2. For each radio, channel and Tx power, same as the display option discussed above.
3. A number of utilization figures from the ARM algorithms on the AP:
   1. Clear is % of the time something was detected on the channel.
   2. Int (interference) is the % of the time that non-Wi-Fi interference was measured.
   3. Tx (transmit) is the % of the time this AP was transmitting Wi-Fi frames.
   4. Rx (receive) is the % of the time this AP was receiving Wi-Fi frames (whether or not they were addressed to it).
   5. (b), (c) and (d) should add up to (a). 100-(a) gives the % of time the medium is un-occupied and is a measure of headroom.

It’s useful to dynamically update this information, to see current information (only a few minutes old, anyway). But this requires us to periodically poll Airwave for the full AP list, something we don’t normally do because it is time-consuming and takes some real-time. So there’s asetting (**Poll for AP Details**) that causes a new poll every (**Location Update Int**) seconds. Switch this on if you want the dynamic updating of ARM information. Because it causes problems with the heatmap calculation, we don’t update the AP list if the new list has a different length from the old one. Broadly, if you have an incorrect floorplan or the AP overlay is wrong or it just looks as though things are confused, exit and re-start Aruba Utilities.

When an AP is isolated like this, touching it again will return the display to the universal heatmap.

The Site Survey feature allows mapping of actual, rather than calculated coverage. Press the **Site Survey Mode** button, move to a point where you want to measure coverage and touch the floorplan at that point. The touch event will start a scan for APs, and the results (Timestamp, location xy and other location estimates, audible AP BSSIDs and RSSIs in dBm) for all audible APs will be logged. On the screen, each reading results in a red square with text showing the bese coverage level in dBm, and the number of APs audible with signal strength equal to at least the yellow threshold on the Handover tab (nominally -68dBm but will vary somewhat depending on the phone’s calibration). When all desired points have been measured, press **Finish Survey**. The points will disappear, but can be recalled with the **Show Survey Results** button.

To get the best results from Site Survey mode, use settings to enable **Filter Scan Results** and disable **Show AP Details** and **Display Heatmap**. This is for clarity, there is no requirement in the code.

To save and process Site Survey results, either save, export and parse the Log file, or take a snapshot of the screen (volume-down + power buttons on phones after Android 4.0). Any suggestions about presenting site survey results will be welcomed.

AMC TAB

The AMC tab shows the data sent to AirWave from an Android implementation of the AirWave Management Client. This is an existing AirWave capability, where client software on a Windows PC scans its Wi-Fi environment, sends reports to AirWave and receives information about the APs it discovered. Much of the value of AMC is the data that is available to the AirWave administrator, device-centric rather than network-centric.

AMC uses a separate account in AirWave (often ‘client’) that must be set up explicitly for AMC. Cookie login credentials are entered as menu items for username and password. The ArubaUtilities default is username ‘client’, password ‘client’. The AirWave **Domain** is used as the address for AMC reports. Update intervals are set at ~6 minutes. The function is enabled with menu item **AMC Client Reports**. The AMC API is JSON over https.

The AMC tab shows data from a variety of sources on the device. This represents most of the information sent in AMC reports. It is all readily obtained on the device using Android APIs. A brief list of each field:

* Current SSID
* Device IP address
* Device MAC address, the MAC of the Wi-Fi interface
* Channel of associated AP
* Signal Strength (RSSI) dBm
* Current link speed (modulation rate) Mbps
* Current associated BSSID
* Authentication and encryption protocols in use, along with authentication status
* The bandwidth and latency figures are from http GETs and POSTs to the AirWave server. The test is run for every AMC report, or manually. The **run test** button kicks off this test: be patient, it can take a minute to run.
* Lat and Long from GPS
* Device Manufacturer and Model
* OS is always ‘Android’ with the rev, e.g. 4.0.3
* For phone devices, the ‘line 1’ phone number
* For phone devices, the current network operator
* The list of neighbouring APs is the same as on the HANDOVER tab except that it is cumulative, and the AMC response classifies each BSSID as ‘known’, e.g in the AirWave database, ‘unknown’ or ‘rogue’.

APERF TAB

APERF is an Android-wrapped version of the familiar open-source Iperf utility <http://iperf.sourceforge.net/> . It is included in ArubaUtilities because engineers often want to run Iperf tests as part of a walkaround site survey, and it is convenient to have it in the same package. Since the Iperf utility itself is unchanged, the familiar command syntax can be used. The IP address used on the command line can be configured with menu item **APerf Host IP Addr**, then it will appear on the command line next time Aruba Utilities is started.

Results can be displayed in compact or verbose format. The latter is almost the same as would be seen on a PC, while the former gives single-line reports.

When Iperf reports results, they are also displayed on the HANDOVER tab in the histogram section with uplink and downlink shown separately.

While client mode works well, we have intermittent trouble when the APERF utility is put in server mode, or one of the bidirectional tests is used – sending downstream traffic to the device sometimes fails. We are working to fix this.

LOG TAB

The log tab creates a text log with timestamped entries for each event. At present, events are scans (shown in the HANDOVER tab), and location fixes such as GPS. The log can be saved to a file on the device using the main menu **Save Log** button. Files are saved to the device’s SD in the D:\Android\data\com.arubanetworks.arubautilities\files folder with a datestamped filename.

Files can be transferred to a PC via USB umbilical – we can enhance to send via email if there’s a need. Reports can be manipulated by any text editor.

This function allows coverage reports to be generated, based on the signal strength of APs scanned at intervals. The most difficult step is to associate each report with a location – at present we use timestamps, we may add various location fixes to each log item if there’s a demand - let us know what would be useful.

Installing ArubaUtilities

There are several ways to install this application:

* As of 31 August 2012, Aruba Utilities is published on Google Play.
* The arubautilities.apk file can be emailed to the Android device as an attachment (according to Google docs, this only works if you are using the Gmail app on the device and have ‘**Unknown Sources’** enabled in settings) and double-clicked.
* It can be transferred via USB umbilical or microSD memory card, but then an installer app must be used to install it.
* It can be posted on a web page and clicked to download and install.
* Or it can be introduced using the Android Device Bridge adb utility on a PC.

The device will need ‘**Unknown Sources**’ enabled, under ‘**Applications**’ before Android 4.0. We also **Use GPS Satellites** (‘**Location & Security’** up to Android 4.0) and ‘**Allow Mock Locations’**, under ‘**Applications > Development’** up to Android 4.0, although neither is required for most ArubaUtilities functionality. In tabular form:

Menu Item Up to Android 4.0

Unknown Sources Settings > Applications

Allow Mock Locations Settings > Applications > Development

Use GPS Satellites Settings > Location & Security

Menu Item Android 4.1 and later

Unknown Sources Settings > Security

Allow Mock Locations Settings > Developer options

GPS Satellites Settings > Location services

Privacy Notes

Aruba Utilities reads information from the Android device, including Wi-Fi scans for SSIDs and BSSIDs, GPS readings, phone number if available, and other configuration information. Most of this information is only used internally in the application, chiefly to determine the device’s location through different means.

Information can be saved to a log file and the log file can be copied from the phone (under user control).

Some information is sent to your network’s Airwave server through the AMC client function. This information is readable by someone with an account on your network’s Airwave server.

This application does not transmit personal information anywhere else.

Revision Diary

12-08-30

Published versionName 2\_4 versionCode 18

- changed back min Rev to 2.2 (from 2.3.3) because some Aruba SEs are still on FroYo. Hope it works!

- finally got Aperf working again but –i and –s are still problematic

- added a location tracking line showing historical track of avg GP location

- changed GP sigmafsquared parameter from 100 to 150

- fixed bug in AP BSSID matching algorithm (was sometimes showing 2 associated APs)

- added site survey feature on floorplan view and to the log file

- changed the startup sequence so if it starts up with GPS disabled on the phone, it will not attempt to use gps or set mock location

- made rotating floorplan disabled by default

- added new Aruba Utilities icons

- improved Lat/Long/Bearing handling, new preference menu items…

Outstanding bugs

* Nexus Galaxy failed to connect with this: I/wpa\_supplicant(25009): wlan0: CTRL-EVENT-ASSOC-REJECT status\_code=16
* Iperf -s option works (can bounce a client off it) but no stdout and can’t kill it
* Iperf -i option causes long runs, thousands of reports in just a few seconds. It seems to be due to a negative timestamp on the stdout?
* Iperf -t times are not too accurate (vs logcat timestamps) but more accurate than my PC running WinXP.
* LatLong to xy and vice versa is still a little suspect

12-06-07

Published versionName 2\_3 versionCode 17

- Added new Aruba OUI 6C:F3:7F

- Added option for AP details with channel and tx pwr

- Changed to require Android 2.3.3 (ver 10) or later (for native code NDK)

- Added pinch/zoom and drag functions to Floorplan view

- Aperf still not working, but crashes don’t interfere with the app

12-05-10

Published versionName 2\_2 versionCode 16

- Improvement to avoid out-of-memory crash from AMC report

- Fix for orphaned service crash from new Iperf

- Fix for crash on empty mlastNReadings list which popped up when new AP scans were overlaid on an old floorplan.

- New Iperf is still a bit unstable

12-04-24

Published versionName 2\_1 versionCode 15

- Warning toast when a cookie is received but not added to store

- Fixed racing clock problem with lifecycle(onPause)changes

- Changed heatmap to give different colours for decades of dBm

- Improved GP averaging to moving window of last 5 x,y averaged

- Changed logs to add location estimates to scans and remove AMC BSSIDs

- Improved location accuracy by weighting by signal strength

12-04-03

Published versionName 2\_0 versionCode 14 for Ken to not require GPS.