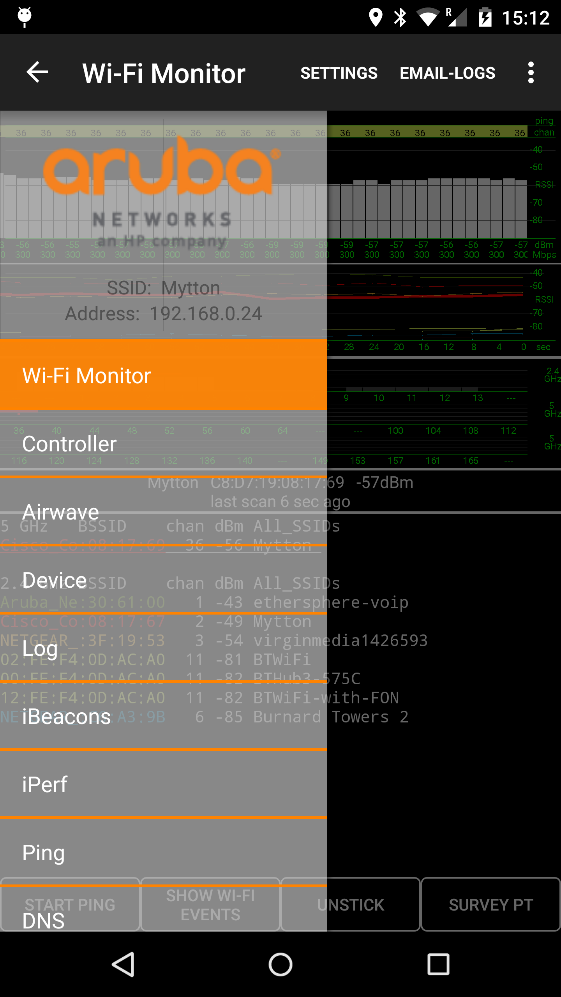
**ARUBA UTILITIES**

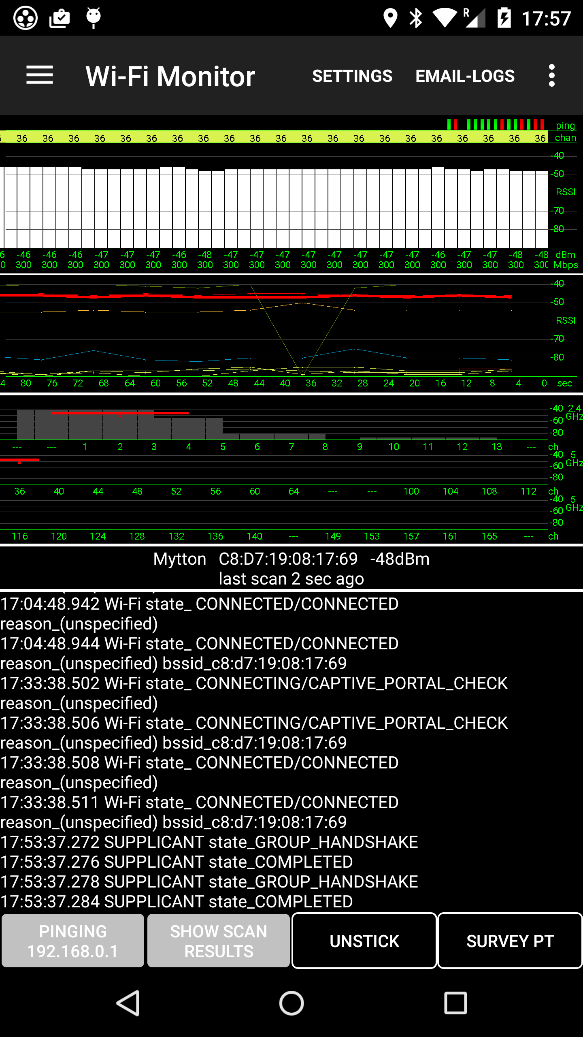
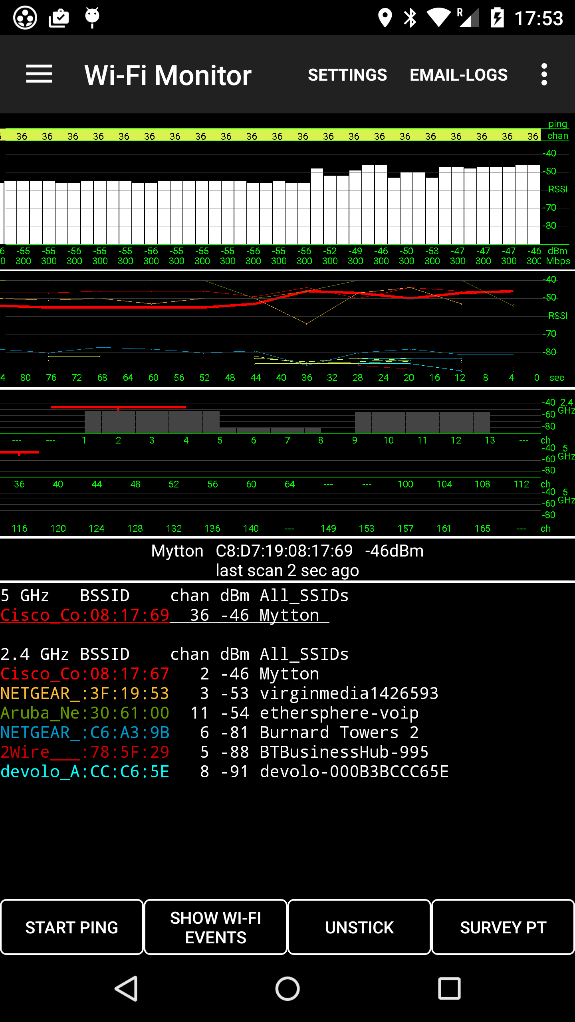
Aruba Utilities includes a number of tools useful for characterizing and troubleshooting wireless LANs from Aruba Networks. Some tools work with any WLAN, others are clients for Aruba’s AirWave management system, Analytics & Location Engine (ALE) and Mobility Controllers. Aruba Utilities includes:

* A Wi-Fi Monitor showing the Wi-Fi environment, including the current access point, dynamic signal strength and RSSI measurements, other access points audible to the device and handover events.
* A Telnet/SSH client works with Aruba mobility controllers, allowing network configuration and monitoring from a mobile platform.
* An AirWave client that downloads the floorplan image and AP details from the 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 AirWave client also offers a locally-generated estimated heatmap and a site survey function that links actual coverage measurements to locations on the floorplan.
* Device information (Wi-Fi, IP, DHCP, cellular status) is displayed along with an implementation of the Airwave Management Client (AMC) that reports device information and scanned APs to your AirWave WLAN management system.
* A Bluetooth Low Energy (BLE) scanner reports nearby iBeacons and other BLE devices with signal strength measurements.
* Android versions the iPerf, Ping, DNS and mDNS offer network test functionality.
* Measurements are written to a plain-text log file and various csv report files that can be emailed for use later.

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 multi-AP WLANS, especially Aruba Networks WLANs.

Each of the Tabs in the application is self-contained:

Wi-Fi Monitor Tab



This Tab shows 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 usually 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 20-90 seconds of RSSI history is shown, depending on screen dimensions. Underneath the RSSI figures the current data rate of the connection (modulation rate) is displayed in Mbps. The figures change colours to give quick indications of quality, the thresholds for colour change are in Settings (**Yellow Threshold, Red Threshold**).

Underneath the histogram view is a historical view (same timescale) of the various APs audible to the device. They are colour-coded to match the BSSID scan list below. The app kicks off a scan (broadcast probe requests on each channel) every 6 seconds (default value, can be changed in Settings **Force Scan Interval**) so new readings are usually at 6 second intervals. Sometimes probe requests or responses are lost in the air, and the graph will drop for one reading.

The next view is a channel-occupancy view. This keys from the associated SSID: all audible APs with that SSID are shown by coloured bars (colours match the BSSID scan list below) and are centred on the AP channel. APs without the associated SSID are shown by grey shadows of appropriate signal strength. Due to limitations in the Android API, channel widths are not available and all APs are assumed for this view to use 20 MHz channels. The top three views on this Tab can be expanded or shrunk in the vertical dimension.

Underneath the channel occupancy view we show the currently associated SSID and BSSID, its signal strength and the time since the last scan.

Next we show the latest scan results. This is populated from probe request/response scan information, a different source of data from the current connection which drives the histogram view.

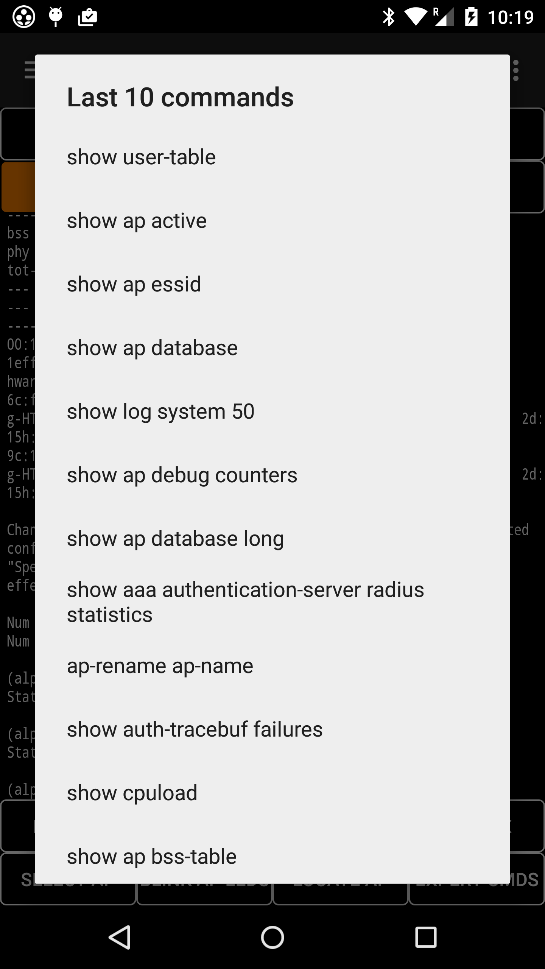
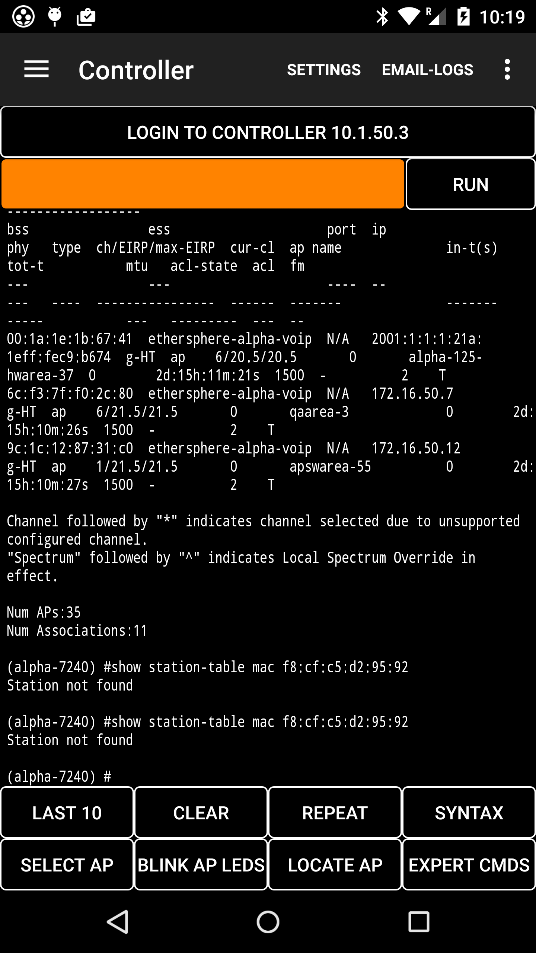
Each scan returns a list, up to 200 items, of BSSIDs with SSID, channel and signal strength (RSSI, dBm). This list can be filtered to only those BSSIDs advertising the currently-associated SSID (Settings item **Filter Scan Results**): it’s sometimes useful to shorten the list in large networks.

In addition to the parameters above, the BSSIDs are shown with manufacturer string for OUI (e.g. **Aruba\_Ne:30:60:00**) and the currently-associated AP is underlined. If the app has been configured to connect to an Aruba mobility controller (on the Controller Tab) or AirWave (the Airwave Tab) then AP names are shown if there’s a match between the BSSID scanned over the air and information from the network (controller or AirWave).

While the Wi-Fi Monitor Tab is set up to key from the currently-associated SSID, it is possible to filter scans on another SSID by entering the name in Settings (**Filter Scan SSID**) and checking **Filter Scan Results**. This can be a useful option if you know which SSID is of interest, but the device can’t associate to it.

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.

Controller Tab



The Controller Tab is a Telnet or SSH client intended for use with Aruba Mobility Controllers. The controller address, username and password are configured in Settings. Other settings are **Allow Controller Login** and **SSH to Controller** (or Telnet).

Start with the **Login to Controller** button which logs in and runs **show ap bss-table** and **show station-table mac**. These also run once in the background whenever the app is started because the BSS information is used to show AP names on the Wi-Fi Monitor Tab.

Type a command in the orange text box and touch the **Run** button. The text box is free-form, and a number of useful commands are packaged in a menu presented by the **Last10** button. The **Last10** button also has the last 10 commands executed. Other buttons are **Clear**, **Repeat** which repeats the command in the orange box every N seconds (long-click for N), and **Syntax** which brings up the Help text.

If the **show ap bss-table** command executed correctly, a long-click on the **Select AP** button will show a menu of all APs by name. Choose one to activate the **Blink AP LEDs** and **Locate AP buttons**. The **Blink AP LEDs** button sends a command to the controller to blink its LEDs, and then allows un-blink. **Locate AP** shows the latest signal strength from the AP, updated every few seconds, to assist with AP hunting. If the AP has several active BSSIDs, the strongest signal is shown.

Expert Commands sends a series of commands to the controller that assist in troubleshooting the client and AP. Expert commands are:

"show station-table mac <client mac>",

"show user-table verbose | include <client mac>",

"show ap debug client-stats <client mac>",

"show ap debug radio-stats ap-name <ap name> radio 0 advanced",

"show ap debug radio-stats ap-name <ap name> radio 1 advanced",

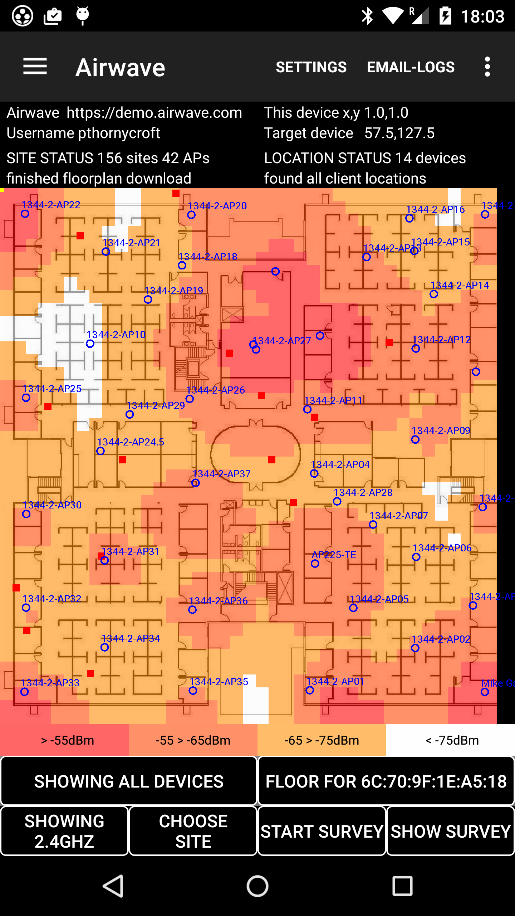
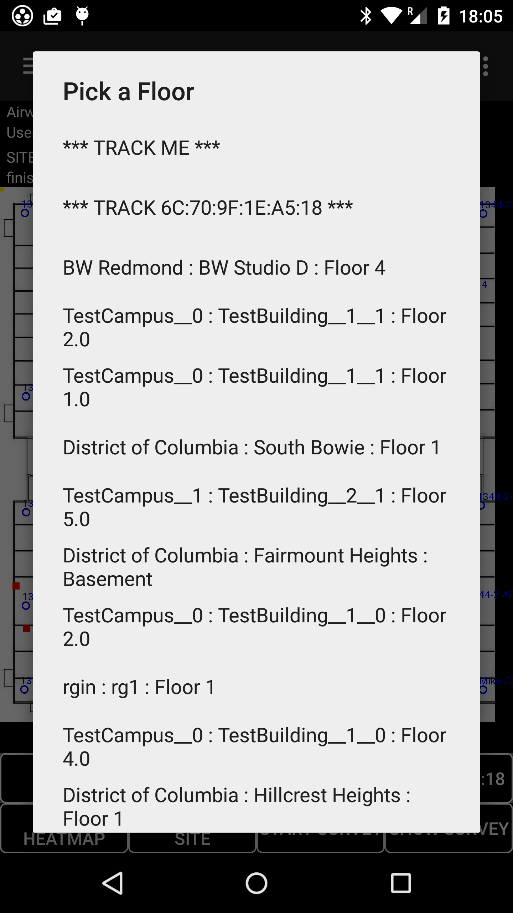
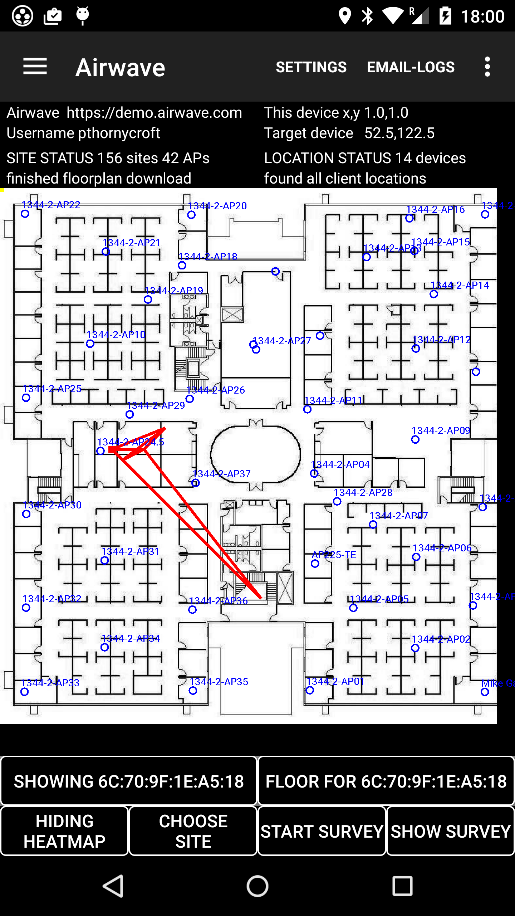
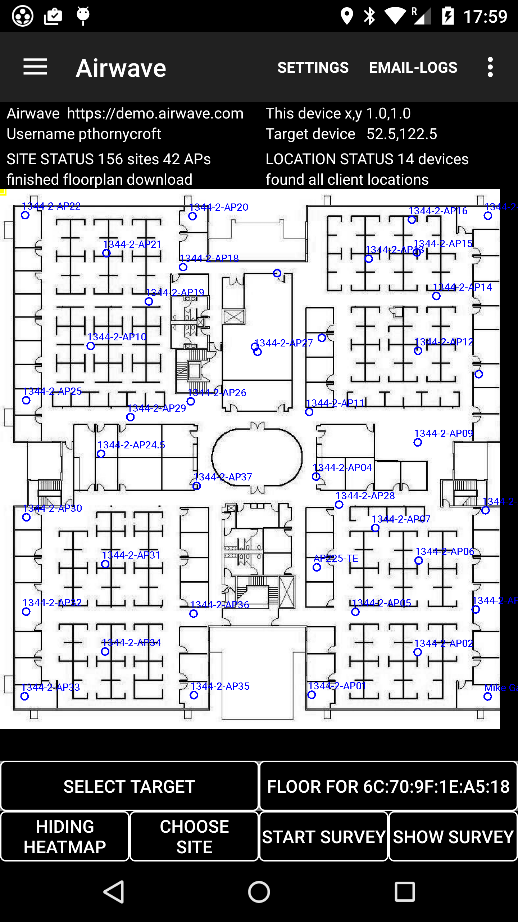
"show wms client <client mac>",

"show ap debug client-table ap-name <ap name> | include <client mac>",

"show ap monitor stats ap-name <ap name> mac <client mac>"

The contents of the large scrolling text view on the Tab are emailed using the email-logs menu button.

Airwave Client Tab



The Airwave Tab shows the WLAN topology, using information obtained from the AirWave AMP and VisualRF services over http. 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.

The AirWave server offers a comprehensive xml/http API through which much of the information seen on the VisualRF screens can be obtained programmatically. For this Tab to function, the AirWave server needs to be accessible from the device and the user must have an account on the server. The device does not need to be associated with an AP on an AirWave floorplan.

Since Rls 7.0, AirWave uses a persistent-cookie http login method over an SSL connection. Aruba Utilities implements SSL (with a ‘trust all certificates’ function, so beware) with an http client. Go to Settings to configure the AirWave IP address or 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 (easily done, especially if AirWave runs on a VM), 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 from Aruba Utilities.

Once AirWave login is successful, Aruba Utilities goes to the top level of the campus – building - floor (site) tree and downloads information about all sites configured on the server. The **Site Status** field will show how many sites were discovered. Use the **Select Site** button to download the desired floorplan image. At this point there should be a number of APs in the **Site Status** field.

The following functions are used in the course of site discovery and floorplan download:

1. Discover all sites using a query of the form <https://demo.airwave.com/visualrf/site.xml> . This returns a list of sites on the AMP server. These are displayed on the **Select Site** button menu with campus\_building\_site names and the site\_id key is bound to the site name.
2. When a site is selected, a query of the form <https://demo.airwave.com/visualrf/site.xml?id=2e298369-4f7c-4732-8b4c-b64ac7a0ec5c&aps=1&images=1> returns a list of APs on the site, along with a key for the floorplan. The AP information is parsed so we can show APs in the right location over the floorplan image. Information includes AP 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 identifier and floorplan identifier, 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 location calculations will be off.
3. The floorplan bitmap is downloaded using <https://demo.airwave.com/visualrf/data/2e298369-4f7c-4732-8b4c-b64ac7a0ec5c/background.jpg> . Most sites in AirWave have a “background” image and “thumbnail” image, the latter is smaller and (at least with older Android devices or when bandwidth is limited) should be used if there is a failure to download the full “background” image due to its size.
4. A lookup to the building identifier 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 origin (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).

Once a floorplan is downloaded, it is added to the Aruba Utilities cache. For offline use, floorplans can be retrieved by long-clicking the **Choose Site** button.

If the above queries are successful, the floorplan will be displayed on the Airwave Tab with superimposed AP information. There are several options for displaying device locations on this floorplan.

The app queries VisualRF for the latest location of its own-device MAC address, and a “target” MAC address if you wish to track another device. The query is of the form <https://demo.airwave.com/visualrf/location.xml?mac=F8:CF:C5:D2:95:92> . The **Select Target** button sets the options, and also has a **show all devices** option.

A Setting for **Show Location Tracks** displays the history of each tracked device.

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

1. From VisualRF (red square, Setting **Show Airwave Location**). 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 measurements (Setting **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, a machine-learning mathematical technique that fits curves to multi-dimensional data, to build a signal strength heatmap. Aruba Utilities 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 GP 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 and takes around 2 minutes depending on the number of APs and the size of the floorplan. The resulting heatmaps for each band can be displayed through the **Display Heatmap** button at the bottom of the screen. GP tracking requires Settings **Show Heatmap** and **Show Locally Calculated Location** and shows the yellow square for location.
4. Android devices have several 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. Settings **Show Location from GPS** switches this on.
5. Android devices include a compass. This can be used to help orient the user, if Settings **Rotating Floorplan** is selected. The floorplan should be rotated appropriately, based on the north\_heading explained above. If no north\_heading 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 technology such as device-based inertial navigation and round-trip-time distance measurements in the future. If this goal can be met, site surveys will no longer require any user intervention to identify waypoints, which most systems must use today.

The Airwave 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 AMON or 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. Really it’s ‘not-clear’.
   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.

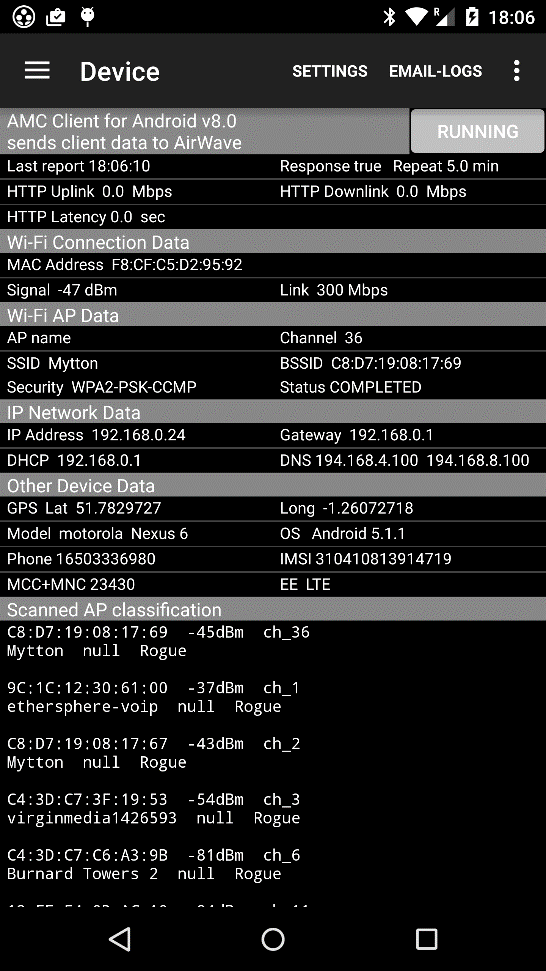
It’s useful to dynamically update this information, to see near-realtime information. But this requires us to periodically poll AirWave for the full AP list, something we don’t normally do because it is time-consuming. So there’s a Setting (**Poll for AP Details**) that causes a new poll every (**Location 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, choose a new Site and then reload the desired Site, or exit and re-start Aruba Utilities.

The **Site Survey** button allows mapping of actual, rather than calculated coverage. Press the **Site Survey** 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 best coverage level in dBm, and the number of APs audible with signal strength equal to at least the yellow threshold on the Wi-Fi Monitor 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 most phones after Android 4.0).

Device Tab



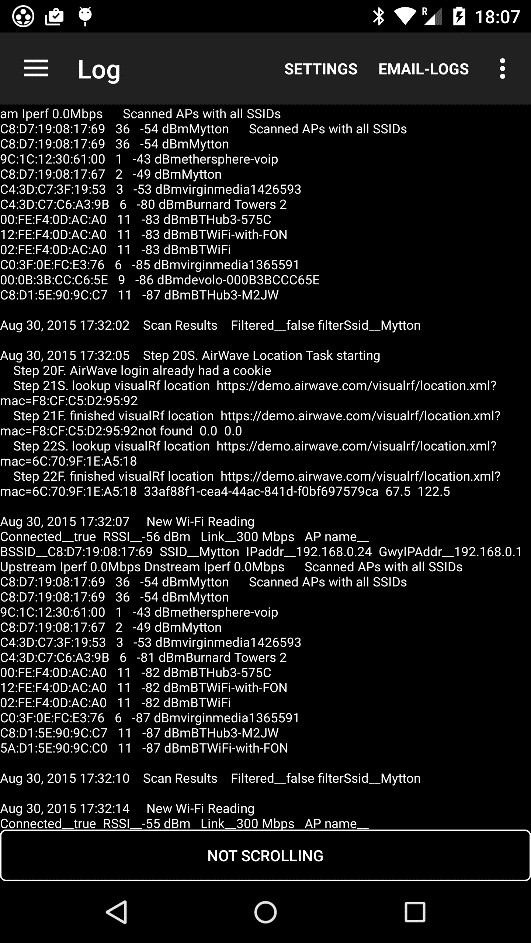
The Device tab shows data about the device, its configuration and status. A brief list of each field:

* The timestamp of the last AMC report, whether the response indicated success (“true” or “false”)
* How often the AMC report is set to repeat (not currently settable, usually ~5 minutes). The Send Report button over-rides this with an immediate report, whether or not the Settings **Send AMC Reports** is set.
* 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.
* Device MAC address, the MAC of the Wi-Fi interface
* Signal strength (RSSI) dBm of associated AP
* Current link speed (modulation rate) Mbps
* AP name (if available from mobility controller or AirWave) for associated AP
* Channel of currently-associated AP
* Current SSID
* Current BSSID
* Security used by associated SSID/BSSID
* Status of authentication supplicant
* IP address, Default Gateway address and DHCP server address
* DNS server addresses (two provided by Android)
* Lat and Long from GPS
* Device Manufacturer and Model
* OS is always ‘Android’ with the rev, e.g. 5.1.1
* For phone devices, the ‘line 1’ phone number
* For phone devices, the IMEI (unique hardware identifier for the device)
* For phone devices, the current network operator’s country code and national code (MCC+MNC)
* For phone devices, the operator name and cellular protocol in use (e.g. EE LTE)
* At the bottom, the list of neighbouring APs is the same as on the Wi-Fi Monitor Tab except that it is cumulative, and the AMC report response classifies each BSSID as ‘known’, e.g. in the AirWave database, ‘unknown’ or ‘rogue’.

The Device Tab includes 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 (type ‘AMC’, named ‘client’) that must be set up explicitly for AMC. The **password** for this account is in Settings, default “client” (but AirWave won’t allow a password that simple, so use something longer like ‘client4AMC’). The AirWave **Domain** is used as the address for AMC reports. Update intervals are set at ~5 minutes. The function is enabled with menu item **AMC Client Reports**. The AMC API is JSON over https.

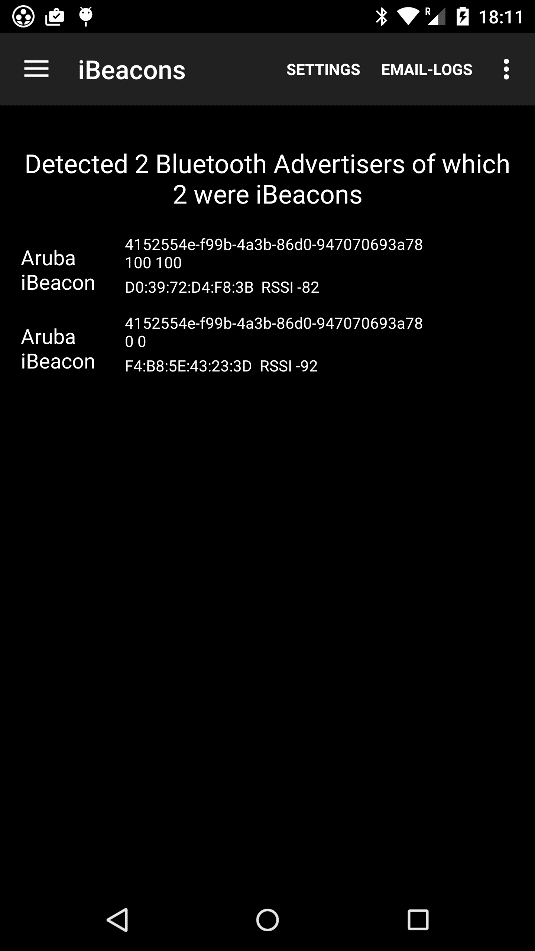
Log Tab



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

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.

iBeacon Tab



Bluetooth Low Energy (BLE) devices such as iBeacons are used extensively for device location and navigation by apps such as Aruba’s Meridian, and it is useful to be able to monitor these beacons to make sure they are working, and are located where the apps expect them.

Most new Android devices (e.g. from Samsung Galaxy S4) have BLE radios that can be used to monitor both Apple iBeacons and non-Apple beacons (e.g. Google Eddystone\*), as well as wearables and other BLE devices.

The iBeacons Tab enables the BLE radio for scans every few seconds and displays all beacons scanned, sorted in signal-strength order. There is a Setting to **Enable Bluetooth Scanning**.

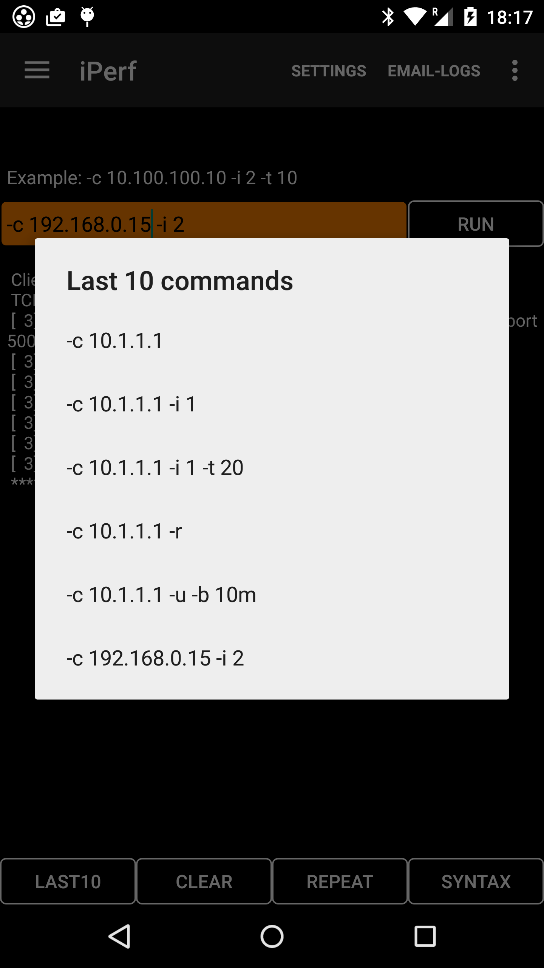
Where a device is an Aruba-manufactured iBeacon, it is noted as such. Special characteristics of the beacon allow this to be determined.

Each beacon has a UUID identifier, Major and Minor index, MAC address and RSSI, and sometimes a name for non-iBeacons.

Touching a device on the list brings up the Detail Tab which increases the scan rate to 2 seconds and shows the RSSI in large font, this is intended for beacon hunting.

\* Eddystone beacons are just treated as BLE beacons for now, the extra Eddystone information is not decoded.

iPerf Tab

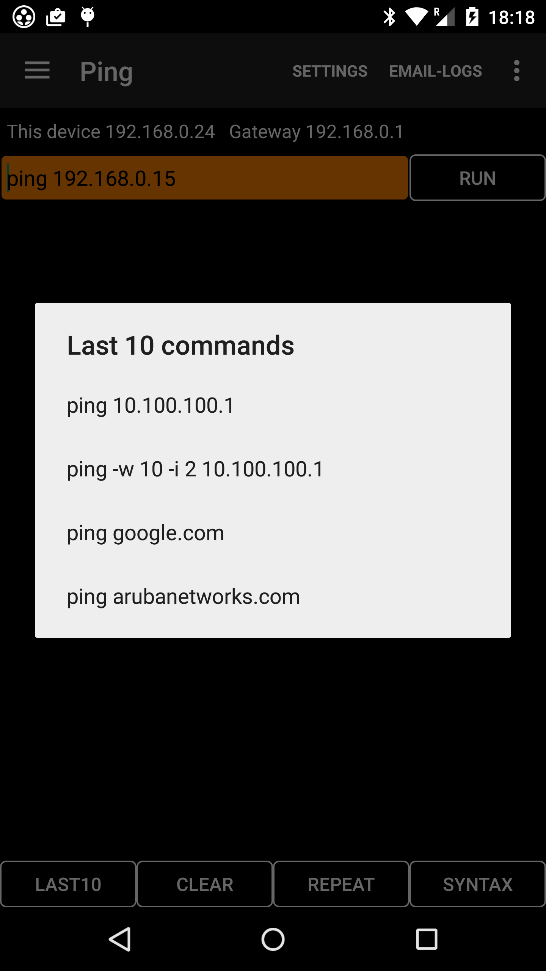
 

This is an Android-wrapped version of the familiar open-source iPerf utility <http://iperf.sourceforge.net/> . It is included in Aruba Utilities because engineers often want to run iPerf tests as part of a walkaround site survey, and it is convenient to have it in the same package. Since the iPerf utility itself is unchanged, the familiar (Linux-version) command syntax can be used. There are no Settings, all commands are set in the orange box. The last command executed is remembered.

For this and other utility tool Tabs, the **Last 10** button shows canned commands and examples of syntax, the **Clear** button clears the display, the **Repeat** button repeats the last command every N seconds (long-click to set N) and the **Syntax** button shows help.

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

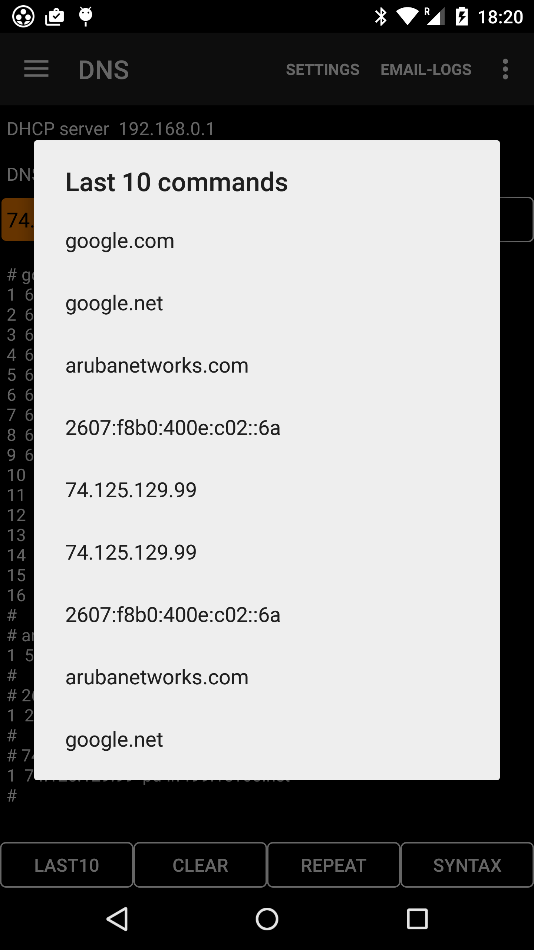
Ping Tab

This Tab gives access to the Linux shell in Android, but not privileged or root access.

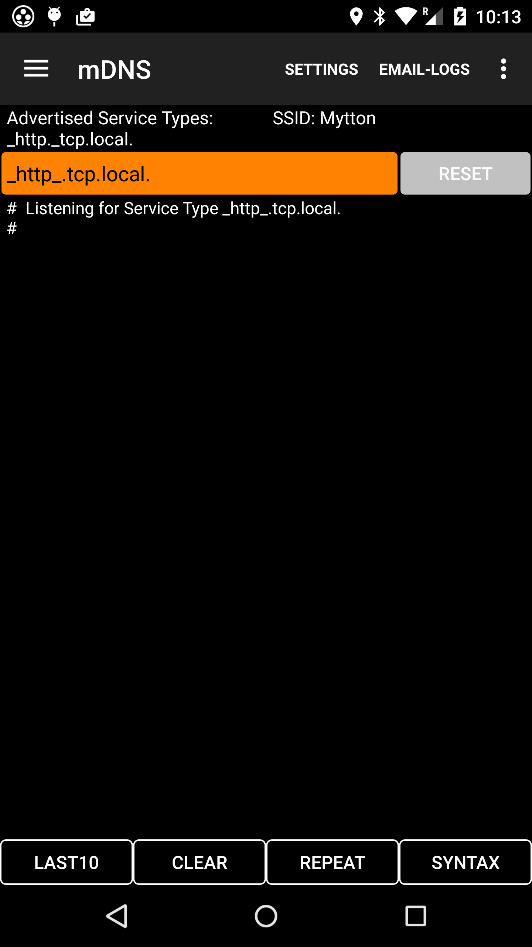
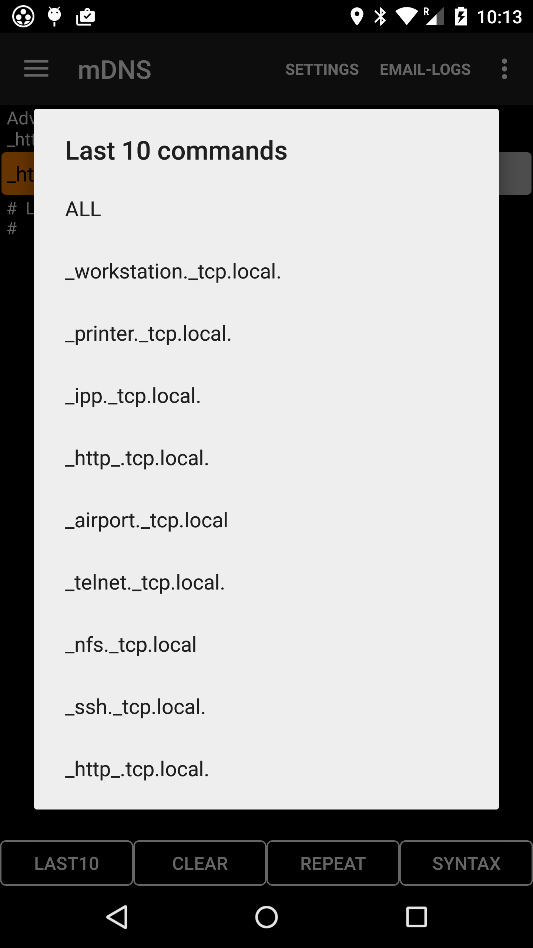
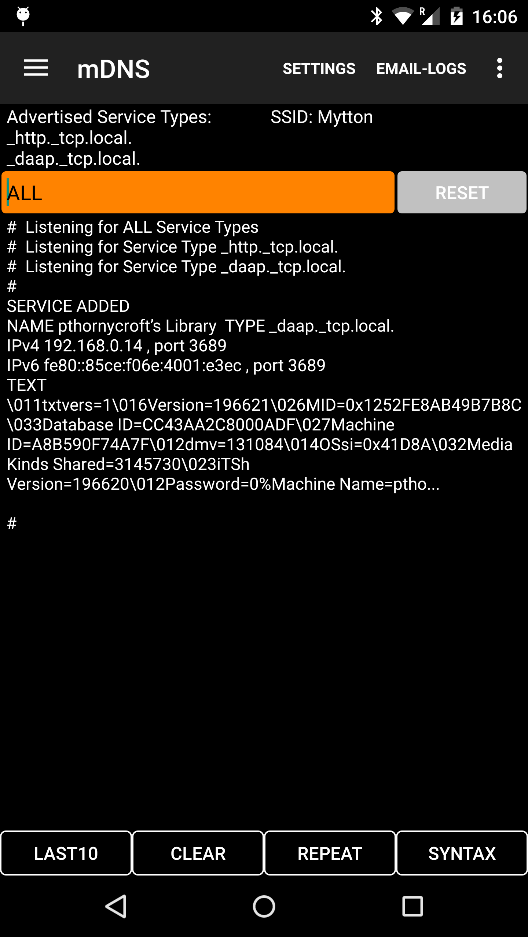
The standard Ping options are available, the **Last 10 button** gives examples.

DNS Tab

DNS uses the servers known to Android (from DHCP or by configuration) for forward or reverse DNS lookups. It is not possible to specify the DNS server used.

mDNS Tab

mDNS is part of Zero-Conf, as used in Apple’s Bonjour for service discovery on L2 networks. Aruba has many AirGroup features that implement and extend aspects of Zero-Conf, it can be useful to monitor what services are available on the network, as seen by a device.

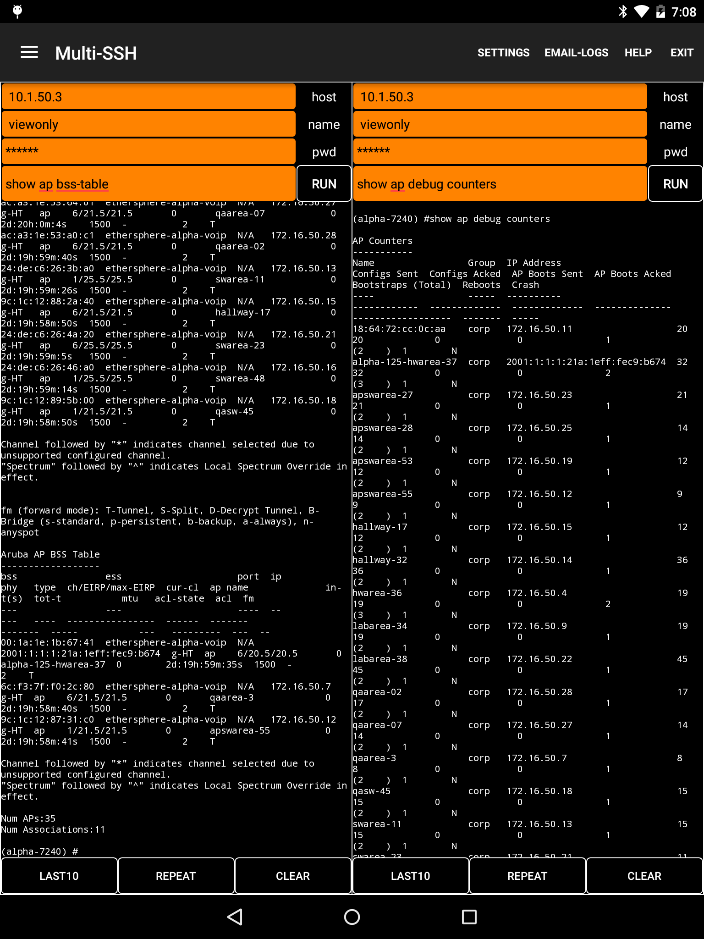
A service-type-discovery listener is active as long as the Tab is open, and discovered service types are listed above the orange box.

To listen for specific device-services, fill in the orange box with **ALL** or a service type of the form **\_http.\_tcp.local.** (note period at end).

The app will start a listener on that service, and all events (services added/resolved and removed) will be displayed.

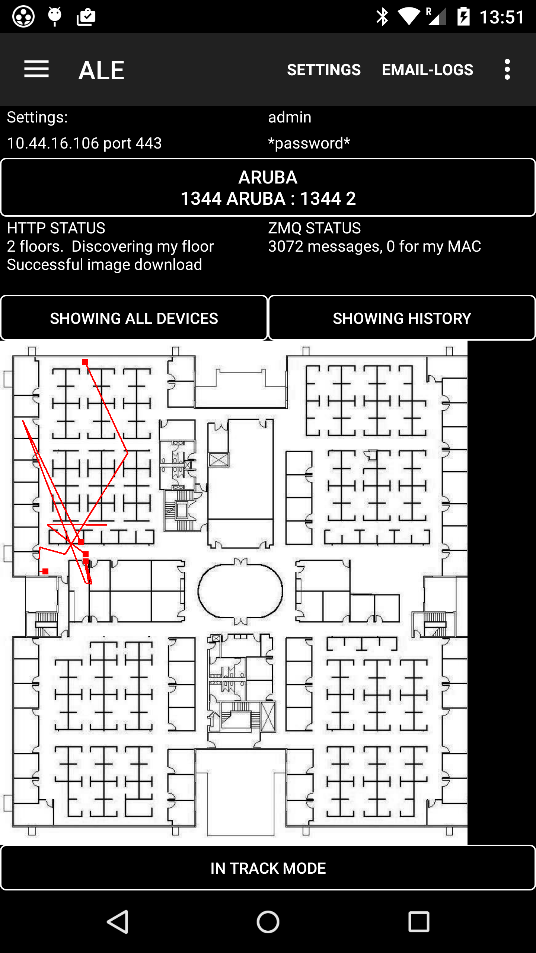
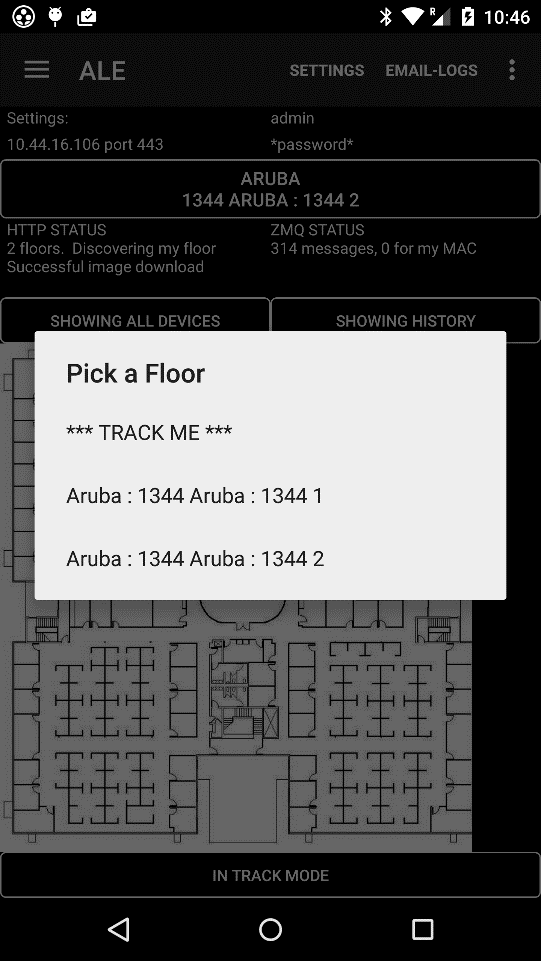
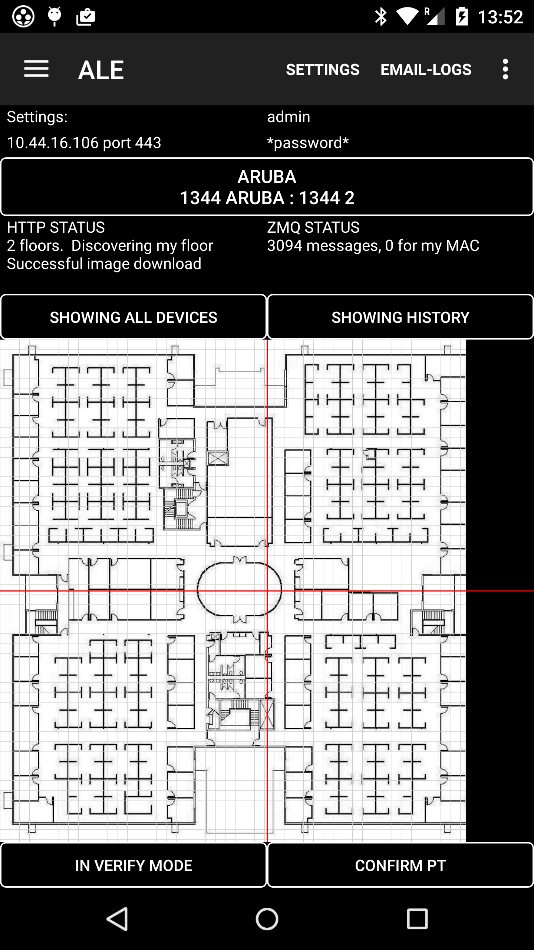
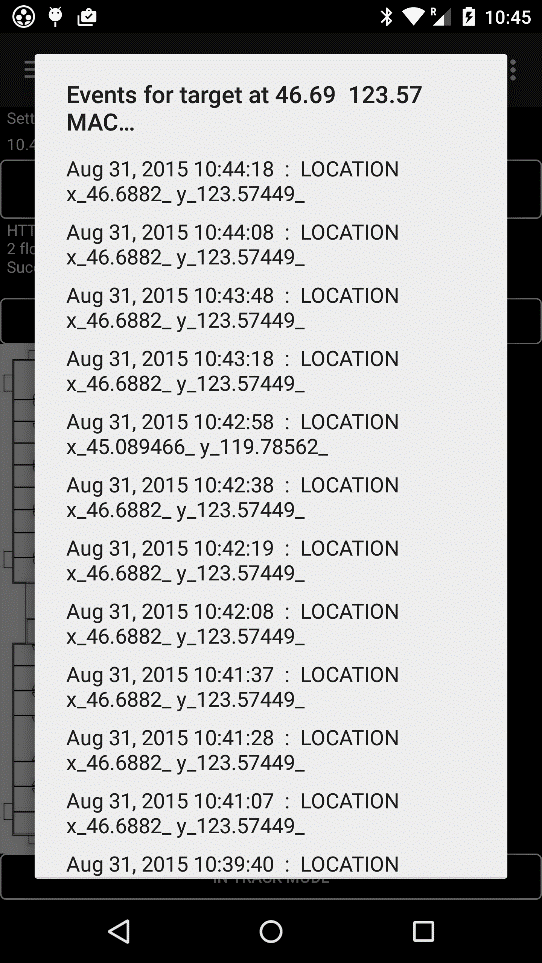
If AirGroup is active, it is usually necessary to listen for a specific service rather than **ALL**, because only when specific service-type listeners are active will the device transmit mDNS requests for AirGroup to recognize and act on.

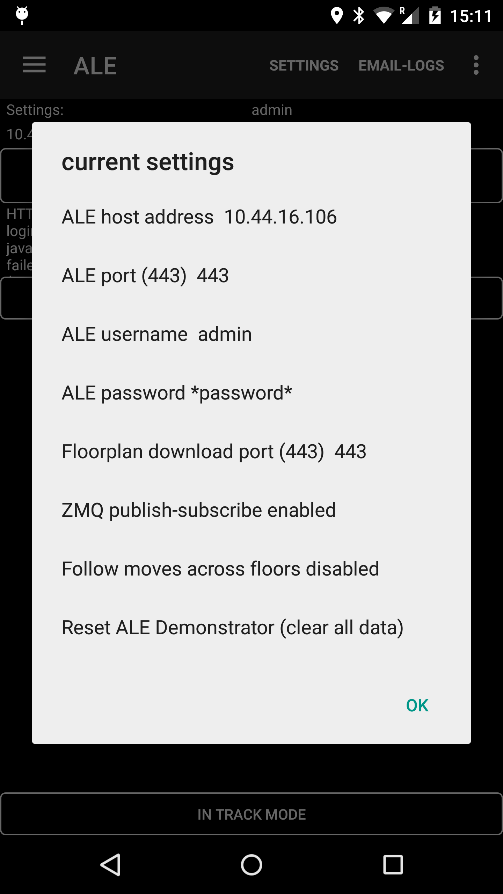
Multi-SSH Tab



For those who like multi-tasking, the Multi-SSH Tab has two independent SSH (not Telnet) clients for logging into mobility controllers. Each has most of the features of the Controller Tab, including **Last 10** and **Repeat** functions.

ALE Tab



This Tab is a client for the Aruba Analytics & Location Engine (ALE). It shows information from the ALE as red target squares over a floorplan image.

The mechanics are very similar to the Airwave Tab, but Settings are configured by touching the top panel for the config screen. Settings include ALE **address**, **ports** and **credentials**.

The ALE client discovers and displays the floorplan image for the “floor” (ALE terminology, equivalent to “site” in AirWave) from the **Select a Floor to Track** button.

Once the floorplan image is in place, the **Showing All** button provides options for tracking a particular device, or all devices on the floorplan.

ALE has an http API for discovering floors and individual device locations, and a publish-subscribe feed using 0MQ (also written as ZeroMQ or ZMQ). When **Showing All**, ZMQ data is used. When showing a particular target or this device, http is also used. With large systems and many targets, the volume of ZMQ traffic can overwhelm an Android device so there is a Setting to **disable** it.

The ALE Tab has a verify function where cross-hairs are shown over the floorplan image. When the **Confirm Pt** button is touched, the current position of the cross-hairs is taken as the true position, and this along with the last reported position from ALE is added to a csv survey file. The file is emailed as an attachment using **Email-Logs**.

Email-Logs and the main menu

The main menu on the action bar has four choices:

* **Settings** brings up a (long) list of settable parameters. The obscure ones are explained in this document. Defaults tend to off/disable, for instance it is necessary to explicitly enable functions such as Controller login, AirWave login, Bluetooth scanning, AMC reports.
* **Email-logs** calls the device’s email functions (you will see a menu to choose from), composes an email and creates a number of attachments (see below).
* **Help** is Tab-specific. On some Tabs the **Syntax** button brings up the help screen.
* **Exit** clears a lot of the app cache before closing the program. In Android, apps are usually put “on ice” in the background when, for instance, another app is brought to the foreground. For Aruba Utilities, we tend to stop activities like scanning when in the background, to avoid draining the battery, but keep data available so there’s less discontinuity when brought back to the foreground. But this can sometimes be confusing when the app is used in a different building or network, when you are moving about it is best to use the **Exit** button and clean out old data.

Attachments for the email-logs function:

(Each file has a timestamp as part of the name. If a file is empty or nearly empty it will not be attached.)

**ArubaUtilitiesLogFile** A text file copy of the Log Tab

**ArubaUtilitiesSurveyFile** If you have been using the Survey Pt function on the Wi-Fi Monitor Tab, this will be a csv (opens with a spreadsheet like Excel) file of that survey

**ArubaUtilitiesControllerTelnetFile** A text file copy of the Controller Tab

**ArubaUtilitiesHandoverViewFile** A snapshot of the Wi-Fi Monitor Tab, taken either when the button was touched (if Wi-Fi Monitor was the active Tab) or last time it ran.

**ArubaUtilitiesCsvScanFile** A csv file of the latest Wi-Fi Scan information, BSSID, SSID, RSSI…

**ALE\_VerifyFile** A csv file with the results of the last ALE verification survey

Installing Aruba Utilities

There are several ways to install this application:

* As of 31 August 2012, Aruba Utilities is published on Google Play.
* The arubautilities\_NNN.apk file can be emailed to the Android device as an attachment 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) under ‘**Applications > Development’** up to Android 4.0, although this is not required for most Aruba Utilities functionality.

Menu Item Up to Android 4.0

Unknown Sources Settings > Applications

Use GPS Satellites Settings > Location & Security

Menu Item Android 4.1 and later

Unknown Sources Settings > Security

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.