Background

A bit of background information should be helpful in understanding the OrangeMesh software. However, this document assumes you are familiar with the concepts behind OrangeMesh, such as what a node is. Please see our user manual or quick start guide for more background information.

OrangeMesh came into being to facilitate the creation of community wireless networks. OrangeMesh is geared towards community wireless networks and is designed to be usable by semi-technical users.

OrangeMesh makes use of several existing open source tools in order to accomplish its mission.

- BATMAN (mesh routing protocol). Maintained by open-mesh.net. (open-mesh.net/batman)
- ROBIN (mesh firmware). Maintained by Antonio Anselmi. (www.blogin.it)
- Open-Mesh Dashboard (existing mesh network management tool). Maintained by Mike Burmeister-Brown of NetEquality. (www.open-mesh.com) Note: this is maintained separately from the BATMAN protocol and is a separate project; the similarity of name is coincidental.

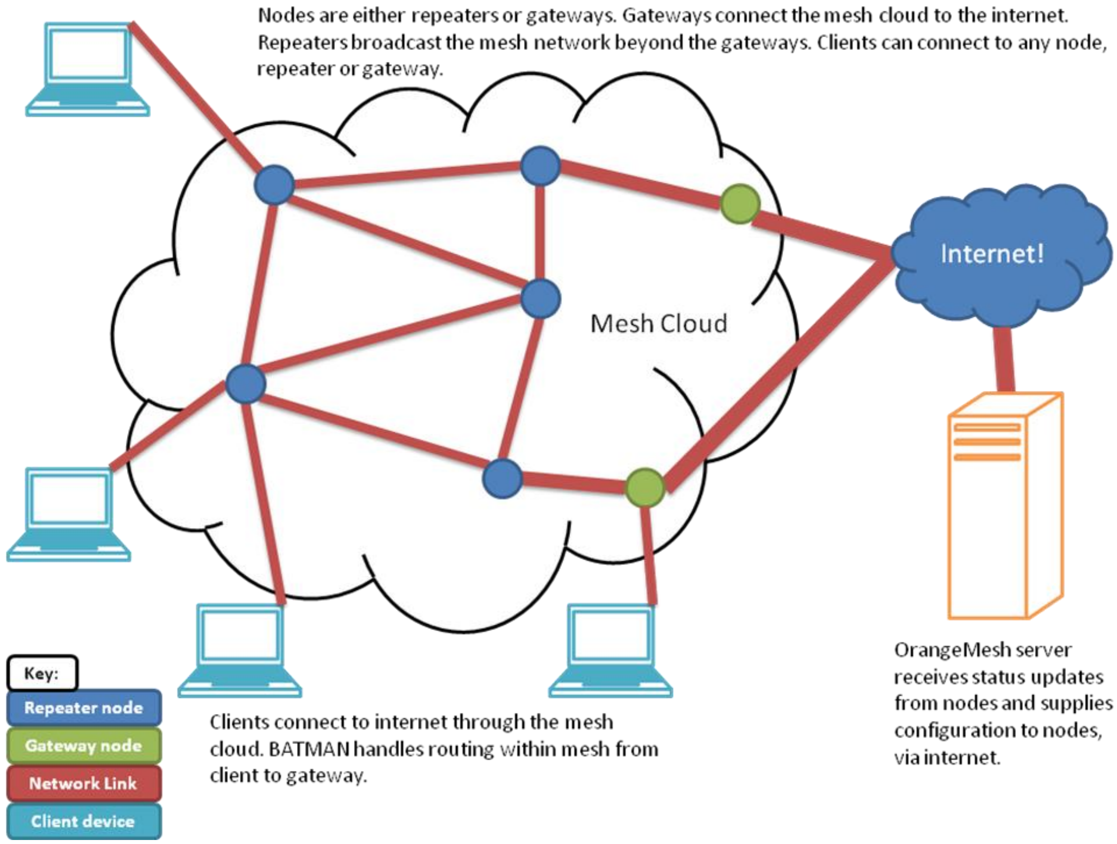
Our project is most closely associated with the Open-Mesh dashboard. We are working closely with Open-Mesh to implement new features and extend the existing dashboard in order to meet the needs of our local client, Orange Networking.

Underlying OrangeMesh is the ROBIN project, maintained by Antonio Anselmi (www.blogin.it). ROBIN is an open source mesh firmware built on top of OpenWRT "Kamikaze", a Linux-based operating system for wireless routers. BATMAN (Better Approach to Mobile Ad-Hoc Networking), the successor to the very popular OSLR mesh protocol used by community wireless groups around the world, is the mesh routing protocol used by ROBIN.

OrangeMesh is a network management dashboard for ROBIN based wireless mesh networks. OrangeMesh allows network administrators to configure all nodes in their networks through a simple web interface, and provides useful ways to visualize network status, such as overlaying node status onto a map. In addition, OrangeMesh allows network administrators to add useful meta-data to their nodes, such as node owner name, email, phone number, and address.

OrangeMesh is a piece of server-side software. It is written primarily in PHP, although there is some Javascript (mostly for generating map views using the Google Maps API), CSS (for skinning), and, of course, HTML. One central OrangeMesh server can support multiple ROBIN networks, each associated with a different network account. OrangeMesh exists completely separately from the mesh network, with all communication between the mesh nodes and the management server done via the internet.

Dashboard Interaction Architecture



The dashboard communicates with the cloud of mesh network nodes to receive status updates and send configuration information.

The mesh nodes individually check in with the dashboard server on a regular basis. They provide the dashboard server node status information, such as uptime, bandwidth used, and user connected. This data is stored in the server's database. The nodes then receive configuration information from the dashboard server based on the network settings that the user has set in the dashboard. The nodes call a script on the server, checkin-batman.php, which handles this exchange of information. Note that all this functionality is defined by ROBIN. Check the ROBIN documentation (http://www.blogin.it/howtorobin) for the most up-to-date information as to the specifics of this interaction.

Migration between servers is handled by separate export and import scripts. For information, see the section on migration below.

User Types

Each OrangeMesh community consists of three main user classes:

1. Visitor – Guests to the community who simply need internet access and do not have any long-term commitments to the maintenance of the network. These people will probably not ever

interact with OrangeMesh, but instead will simply use a ROBIN mesh network for internet access.

2. Neighbors (User) – These are members of the community with a vested interest in the network, perhaps because they rely upon their community's mesh network for internet access. They primarily just want internet access, but may be willing to take a greater role in maintaining the network by hosting a node in their own living space. The neighbor also wants to be able to diagnose trivial network problems on their own without administrator assistance, and as such will probably use OrangeMesh to view network status information.

3. Geeks (Administrator) – Technical or semi-technical community volunteers or workers who have taken on the responsibility of maintaining a ROBIN mesh network. These people will be the primary users of OrangeMesh, and want to see network status information on a very detailed level, as well as a general overview. Further, they want to be able to control as many aspects as possible of their mesh, including managing which nodes are part of the network and network configuration. These are probably the only users who are willing to spend time reading a manual.

User Scenarios

Vinny the Visitor. Vinny is in town for the weekend visiting his cousin Billy. After long hours of driving, he finally arrives at his cousin's house and decides to check his email. Billy's not the most technically inclined fellow, and does not have broadband internet access at his home. Vinny turns on his laptop, notices the open wireless network called "OrangeMesh Community Wifi", connects, and opens up his browser to navigate to his webmail. He is greeted by the network's splash screen, and is delighted by the friendly, inviting page that welcomes him to the community and displays terms of use. He clicks the big button at the bottom that says "Connect" and is forwarded to the webpage that he wants to see.

Nelly the Neighbor. Nelly works during the day and uses the computer during her downtime in the evenings. Mostly she likes to send pictures to her sister, do some online banking and pay her bills, and read up on her favorite hobby, live-action Connect Four. She's a very generous person, and so she shares her expensive broadband connection with her neighbors by hosting an OrangeMesh node in her house. She did this by purchasing a ROBIN node and plugging it into her existing home router. She doesn't really know what the node does, but she is curious about the network, and sometimes she logs into the dashboard just to see how many people are connected to the network and click on the pretty widgets that represent nodes to look at the exciting AJAX drop-down effects. She doesn't use the dashboard very often, but she did register with the network as a neighbor so she could have some control over what information about her is attached to her node.

Garth the Geek. Garth is a 17 year old high school student who narrowly avoided becoming disillusioned by the meaninglessness of his suburban existence by getting involved with his community wireless network. Garth is no networking professional, but he grew up with computers and is comfortable using them. He's the kid his teachers go to when they can't figure out how to use the DVD player on the TV or don't know how to hook their laptop up to the projector. Garth is also on the Geek Team for the OrangeMesh Community Network, and after coming home from school he gets an email from a frustrated community member that their internet connection is very slow. He logs into the dashboard, and views the network status list to analyze the network. He notices that the number of hops to the gateway for one part of the network is unusually high, and upon further investigation notices that the gateway in that part of the neighborhood has been down for three hours. He clicks on the entry for the node to find out the contact information for the node owner, and gives them a call. The node owner realizes that her husband unplugged the node so he could plug in his new remote controlled foam dart launcher. After chastising him, the node owner plugs in the node, and Garth verifies on the dashboard that the node is back online and functioning properly.

Use Cases

Administrator

Network Creation. The administrator navigates to the dashboard server in their browser. He/she selects “create network” from the menu. The administrator enters the desired network account name and password (twice, to verify), as well as a network location (either postal code, or city/state, or some other descriptive location) and email address. The administrator selects “submit”. If the network name is taken, the administrator receives an error message (“The network name you selected has already been taken.”), and is redirected back to the network creation page. If the passwords do not match, a similar error is displayed and the administrator is directed back to the network creation page. If the network creation is successful, the administrator is logged in as an administrator and redirected to the settings page.

Changing Network Settings. This page is a form where the administrator will be able to change the network account settings, which are network password, primary email address for network, and additional addresses to send network status updates. The administrator will be able to set network settings on both network interfaces of the nodes. For AP1, these settings are ESSID, network key, splash page settings, throttling, and access control. For AP2, these settings are enable/disable, ESSID and network key. The administrator can also configure advanced options, which are the node root password, interface isolation, enable usage of test firmware, and enable/disable of migration features. When the administrator has made the changes he/she desires, he/she selects “submit” and the changes are written to the database. The settings page is then refreshed and the new settings are reflected in the settings page.

Adding Nodes. The administrator is able to click on a map to add a node. A info box appears where the administrator clicks and prompts the administrator for a node name, MAC address, and description. The administrator may optionally add node owner information. When the administrator clicks “add”, the node is added to the network as an “activated” node, and a marker is added to the map.

Editing Nodes. The administrator can click on a node marker in the “Add/Edit Nodes” map. A two-tab info window will appear. One window contains basic node information (such as name and MAC address), as well as a “delete node button”. When the delete node button is pressed, the node marker is removed from the map and the node is “deactivated”. When the administrator selectes the “update” button, any changes made to the form are written to the database. The same is true for the other tab, which shows node owner information. The administrator may drag any node to a new location by clicking on the node and dragging it.

Viewing node status on a map. The node map displays node basic node status information. Markers are green if the node is running, and grey if the node is down. A number appears on the marker to indicate the number of users connected to that node. Clicking on a node marker opens two info window, one containing node status information such as last checkin, bandwidth transferred, and users connected. The other window shows node owner information. None of this information is editable.

Node Status in a list. This is a listing of all activated nodes in the network. It displays a node in red if it is down, and in black if it is up. If a node is a gateway, its name is written in bold. The list is sortable and contains all node status information, including users connected, bandwidth transferred, hops to nearest gateway, and last checkin time. If no nodes are associated with a network, a message stating so is displayed and the administrator is directed to add nodes.

Node Owner Information in a list. This list shows the node owner information for every node in the network. This includes the owner name, email, phone, and address. The last column shows the node's activation status, either Pending, Activated, or Deactivated. The administrator can click on a node name to change node owner information, including activation status.

Login. The administrator is prompted for a network account name and a password. If the network does not exist, or if the password is incorrect, the administrator receives an error message and is prompted to try again.

Logout. Clears the currently logged in session and redirects back to the home page.

User (Neighbor).

Adding Nodes. The user is able to click on a map to add a node. A info box appears where the user clicks and prompts the administrator for a node name, MAC address, and description. The user is required to include node owner information. When the user clicks "add", the node is added to the network as an "pending" node, and a marker is added to the map.

View My Nodes. The user may search for nodes by email address. This displays basic node status information as well as the pending/activated/deactivated status of nodes. It will mark nodes that are down in red, but nodes that are up in black. If no nodes match the input email address, an error message is displayed.

Node Status in a list. This is a listing of all activated nodes in the network. It displays a node in red if it is down, and in black if it is up. If a node is a gateway, its name is written in bold. The list is sortable and contains all node status information, including users connected, bandwidth transferred, hops to nearest gateway, and last checkin time. If no nodes are associated with a network, a message stating so is displayed and the user is directed to add a node.

Viewing node status on a map. The node map displays node basic node status information. Markers are green if the node is running, and grey if the node is down. A number appears on the marker to indicate the number of users connected to that node. Clicking on a node marker opens two info window, one containing node status information such as last checkin, bandwidth transferred, and users connected. The other window shows node owner information. None of this information is editable.

Detail for Features not Derived from Open-Mesh

Migration between Remote and Local Dashboard Servers

Motivation: Users of the dashboard should have the **freedom to choose from where they manage their dashboard**, be it a remotely hosted solution from NetEquality or their own locally hosted server.

Specification: Under the "Advanced" pane of the "Edit Network" screen, the administrator will be able to export their information to a remote dashboard server, or enable their server to receive data from remote dashboard servers. The first of these is "Export" functionality, while the second is "Import".

To export the network information stored in the database, the administrator will have a dialog to enter the IP address or hostname of the remote dashboard server. The information tip located to the right of this dialog (See existing Open-Mesh configure network page) will remind administrators how to find this information. It will also remind them that this operation will send all their data to a remote server and will provide a link to more extensive documentation. It will finally remind them that the import capability on the remote server must be enabled for the transfer to work properly (see below). After the administrator puts the remote dashboard server's information in, they will click "Export" and the export function will begin. If the remote server does not have importing enabled (see below), the administrator will receive a "failed transfer" error message. This message will also appear if the remote server is not reachable. Otherwise, the administrator will be told that the transfer is in progress and will be notified when the transfer is complete.

The Import functionality will be mostly transparent to the administrator. Due to the security risk of allowing an arbitrary user to send large amounts of data to the dashboard, this functionality will be disabled by default. The administrator must specifically enable Importing if he or she would like to import data. The administrator may turn on importing by clicking the "Enable Import" checkbox in the advanced section of the network configuration page. When they do so, the dashboard will accept imported data until they disable it. They can disable this functionality by Deselecting the checkbox.

Node Owner Information System

Motivation: The nodes in a network may be spread over a geographically disperse area, and will often be housed inside the homes of community members. Network administrators will need access to these nodes at certain points to perform maintenance and troubleshooting. In order to do this, the administrators will need **records of who owns each node**, along with a way to contact them.

Secondly, in order to increase community involvement and more easily expand the network, community members should be able to **apply to become node owners**. Allowing non-administrators this ability will distribute the workload of initial setup and allow organic expansion down the road

Specification: This functionality has two components, one publicly accessible and the other privately accessible.

The publicly accessible component will be two new items in the network status page, one entitled “Add Nodes to Network” and the other “View My Nodes”. The user may click “Add Node” to bring up the public apply to add nodes interface. This page will be identical to the “Add Node” interface given to administrators. The user selects the location of his/her node on the map, and a dialog box appears where the user has clicked. The user is prompted for the MAC address of the node, a node name, a node description, the user’s name, email address, phone number, and physical address. The new node will be added to the database, and its approval status will be set to “Pending” awaiting action from the administrator. Then the user will receive a message saying “Node information sent to administrator for approval” if the process was successful. If it was not (i.e., node already existed in database) the user will receive an error message.

All fields are required fields. If the user does not enter the required information, then the user will receive an error message stating they did not fill in the form correctly; fields with errors will be indicated. This error condition will also be triggered if they do not enter information in the proper format (i.e., an invalid MAC address). If a user tries to add a node to the network that has already been added, the user will receive a failure message stating that the node they tried to add already existed.

The “View My Nodes” feature will prompt the user for their email address. The dashboard will search its database for all nodes that match that email address and will display certain vital statistics about them: Uptime/Downtime, Approval Status (Pending/Activated/Deactivated), and bandwidth usage. If the user enters an email address that does not correspond to any nodes, they will receive an error message stating that there are no nodes linked to the email address they entered. This concludes the publicly accessible view of the node information system.

The privately accessible component is only available to administrators. They will have a button in the menu that reads “Node Info List”. All these pages will show a listing of all nodes that have the corresponding status, including the following information: Node Name, MAC address, the Node Owner name, email, phone, and address, and activation status.

If no nodes match (i.e., there are no pending or rejected nodes), a message will display in place of the node listing.

The email address for each node owner will be a clickable link. The entire table will be sortable by every column, as well.

The Administrator can also view all this information in the Google Map interface. The node information will display in popups over each node, similar to what currently exists in the network status page in Open-Mesh. Each node will be color coded based on its status.

Requirements Summary

First Priority

- Network account creation
- Administrator login system
- Store Network Statistics from node checkins in database:
  - Bandwidth Usage (per user, per node, and overall network)
  - Node uptime/downtime
  - Users connected (per node, overall network)
  - Node neighbors and signal strength
  - Hops to gateway (per node)
  - Number of gateway nodes in network
- Network configuration
  - Web interface to write node configuration statistics to database
  - Account settings configuration
- Status list for node information
- Node management
  - Add node form interface
  - Link node owner contact information to nodes
- Input sanitization (prevent SQL/HTML injection)
- Checkin script
  - All node status information
  - Basic configuration (non-captive portal)

Second Priority

- Node Management
  - Map-based add/remove/edit node from network (replaces form based interface)
  - Public “My Nodes” listing – view nodes that match an owner’s email address
  - Administrator node owner information edit interface
- Visualization
  - Google Map-based physical network layout visualization
  - Visual trouble indicators – map marker colors, color coding in list
- CSS Skin for dashboard
- Migration
  - Export data from network
  - Import data from network

Third Priority

- Checkin script
  - Captive portal support (suspended pending updates to firmware)
- Visualization
  - Google Map-based visualization of links between nodes
  - 24hr node status image
- Community
  - User-configurable node owner contact info
    - Privacy options: Display all contact info, phone # only, or obfuscated “contact” link
  - Identity verification system: Community members vouch for the identity of new community members by verifying contact information.
- Network Statistics
  - Throughput between any two points in network (update: will require firmware updates)