

Adding Wifi Support to QEMU Simulation in LibreMesh

Project Details:

Adding wifi support to QEMUSimulation in Libremesh

LibreMesh relies on continuous testing to ensure its robustness across various network conditions and hardware setups. Currently, testing new firmware images lacks a standardized method to include Wi-Fi simulations within a virtualized environment. This project aims to add Wi-Fi support to existing QEMU simulations for LibreMesh, allowing near-realistic network testing using virtual Wi-Fi interfaces. By leveraging the `mac80211_hwsim` module and integrating existing OpenWrt-friendly virtual Wi-Fi tools, we can enhance testing automation and reproducibility.

User-visible changes:

In the Virtual WiFi Test Setup, we used the `load_virtual_wifi_module` function from the `setup_virtual_wifi.sh` script to load the `mac80211_hwsim` module, allowing us to simulate WiFi radios without the need for physical hardware.

In the Scripted Testing Framework, functions like `create_vm`, `attach_interface_to_vm`, and `test_connectivity` from the `test_runner.sh` script automate the process of creating QEMU virtual machines, attaching network interfaces, and verifying connectivity between them using `ping`.

To ensure reproducibility, we used functions like `verify_image_checksum`, `reset_environment`, and `download_base_image` from the `reproducible_setup.sh` script, which help maintain a clean and consistent environment for reliable test results every time.

For testing LibreMesh in QEMU, we used the `launch_libremesh_vm` and `forward_ports_to_vm` functions from the `start_libremesh_vm.sh` script to boot the VM with LibreMesh firmware and enable access to the LuCI interface through a browser on the host system.

Test Plan:

Test Plan is required to verify that the virtual Wi-Fi setup using QEMU and mac80211_hwsim works correctly with LibreMesh firmware. Since this project replaces physical testbeds with virtual ones, the entire mesh behavior needs to be validated under simulation. We test the complete setup from virtual interface creation to mesh connectivity and routing protocol behavior.

Following are some of the tests:

- a. Are virtual Wi-Fi radios created properly using mac80211_hwsim?
- b. Can QEMU VMs boot correctly with the LibreMesh firmware image?
- c. Is LuCI accessible using the forward_ports_to_vm setup?
- d. Are mesh nodes visible to each other using bmx6 or ping?
- e. Do routing tables update when nodes are brought down and up?
- f. Are scripts like create_vm, attach_interface_to_vm, and load_virtual_wifi_module working as expected?
- g. Is the environment reproducible every time using reset_environment and download_base_image?

These are some of the key checks. If any of them fails, we debug using dmesg, iw dev, ip link, and LibreMesh logs. For routing checks, we use ip route, ping, and bmx6 -c topology.

To test these phases, we use the scripts:

setup_virtual_wifi.sh → loads mac80211_hwsim, creates interfaces

test_runner.sh → automates VM creation, interface setup, and connectivity tests

start_libremesh_vm.sh → boots VMs with LibreMesh firmware and forwards ports

reproducible_setup.sh → resets and verifies clean environments. Each script's function is tested in isolation, and then all are used together to simulate a full testbed.

Approach:

Current Scenario:

LibreMesh testing often relies on real routers. That's not scalable or affordable. Virtual routers can be set up using QEMU, but they don't support wireless by default. Also, wireless routing protocols can't be tested easily without simulating WiFi.

Aim:

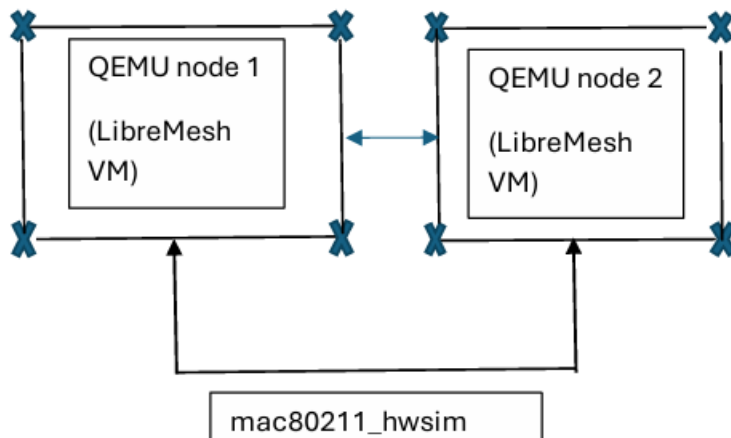
To develop a virtual Wi-Fi testing environment for LibreMesh by integrating mac80211_hwsim with QEMU simulations, enabling automated, reproducible, and near-realistic testing of Wi-Fi-based mesh networks without physical hardware. Proposed

Method:

This project aims to build a virtual environment to test LibreMesh routers without needing real hardware. We will use QEMU (a virtual machine tool) and mac80211_hwsim (a Linux module that creates fake Wi-Fi radios) to simulate wireless mesh routers on a computer.

Each virtual router will run LibreMesh using an OpenWrt image. These routers will be connected to each other through simulated wireless interfaces (wlan0, wlan1, etc.) created by the mac80211_hwsim module. This setup allows us to test wireless mesh protocols (like BMX6 or BATMAN) even if we don't have physical routers.

We will write shell scripts to automate everything: creating virtual routers, configuring wireless connections, starting mesh routing protocols, and testing network connectivity. These scripts will make the process faster, easier, and repeatable.



Here we are using QEMU to create virtual routers that run LibreMesh. Since VMs don't have Wi-Fi, we use the mac80211_hwsim module to simulate wireless radios. Each VM gets a virtual Wi-Fi like wlan0, which connects through the host. This setup lets us test mesh protocols like BMX6 without physical routers. It is useful for easy, low-cost, and repeatable wireless network testing.

- **Download and prepare images again if testbed is broken.**

```
function download_base_image():
```

```
wget http://downloads.openwrt.org/.../libremesh.img
```

```
function reset_environment():
```

```
killall qemu-system-x86_64
```

```
remove temp VMs and logs delete old images (if needed)
```

```
main():
```

```
reset_environment()
```

```
download_base_image()
```

This ensures that all simulations start from the same base image and environment.

Plan

The implementation can be broken down into three milestones:

MILESTONE 1: Initial Research & Basic Setup :

a. Review of existing tools.

- Study mac80211_hwsim, vwifi, and OpenWrt test frameworks to understand how virtual Wi-Fi and router testing work.
- Explore how LibreMesh firmware operates and how routing protocols like BMX6 are tested.
- Understand QEMU basics and how it can be used to simulate LibreMesh routers.

b. Setup of base QEMU and LibreMesh environment

- Create a basic setup where a LibreMesh firmware image boots inside a QEMU virtual machine.

- Ensure SSH or LuCI interface is accessible using a port-forwarding method (forward_ports_to_vm script).
- Verify the VM starts successfully and acts like a normal router.

MILESTONE 2: Virtual Wi-Fi and Mesh Simulation:

a. Integration of mac80211_hwsim module

- Use the setup_virtual_wifi.sh script to load mac80211_hwsim and create fake Wi-Fi interfaces (wlan0, wlan1, etc.).
- Check whether the interfaces appear using iw dev and ip link.

b. Attaching virtual Wi-Fi to VMs

- Write attach_interface_to_vm to link each virtual radio to a VM's network interface.
- Validate the VM can access the interface and recognize it as a Wi-Fi adapter.

c. Multi-VM mesh formation and routing check

- Use the create_vm script to launch multiple VMs and connect them into a mesh.
- Test connectivity using ping and mesh routing protocol tools like bmx6 -c topology.
- Verify routing updates if any node is restarted or removed.

MILESTONE 3: Automation, Reproducibility & Documentation:

a. Script automation and test runner .

- Develop test_runner.sh to automatically run tests like VM setup, interface check, mesh connectivity, etc.
- Write reproducible_setup.sh to reset the testbed and reproduce the same network environment.

b. Integration with LibreMesh tools .

- Ensure compatibility with existing LibreMesh CI testing frameworks and test it across multiple virtual topologies.
- Validate that your testing scripts work with routing protocols and the default LibreMesh environment.

c. Documentation and Finalization .

- Write setup instructions for developers on how to use the test scripts and run mesh simulations.
- Perform final testing and debugging of all scripts.
- Publish blog posts showing the workflow, screenshots, and key results