# OpenWRT based study on QCS mesh network

*Sundararaman Venkatesh*
*Department of Electrical and Computer Engineering,*
*University of Florida*

**Abstract:**

This project involves the study of OpenWRT in a wireless mesh network. OpenWRT is an open-source project to create a free embedded operating system for network devices. The default firmware found in the router is replaced by the OpenWRT firmware by building a serial cable between the computer and router. OLSR (Optimized Link State Routing) protocol is used to enable it to connect to a Mesh Network and act as a part of the Network.

## 1. Introduction:

OpenWRT is a Linux based firmware program which is used to customize embedded devices and residential gateways. It can be installed in routers by building a serial cable connection between the router and the computer. Building a serial cable refers to connecting the computer with the router through necessary serial cables. The firmware that is inbuilt in the router is then replaced by the OpenWRT firmware. This router with the modified firmware can be connected to a mesh network if it has the OLSR protocol embedded in it. The OLSR package is added if required (packages can be added as and when required in Linux) so that it enables the router to connect to the qcsmesh network.

## 2. OpenWRT in Netgear WGT634U:

The NetgearWGT634U Wireless Router has a Broadcom (type 0x29007) CPU, running at 200MHz. This CPU is not compatible with the Pentium (or similar) on a typical desktop computer, i.e. a simple binary program can't be taken and executed in the Netgear router. The operating system on a NetgearWGT634U is Linux and is different from that in a desktop pc. The Linux was specifically compiled for Broadcom CPU. The router has 8 MB flash memory and 32 MB volatile memory. Flash memory is non volatile memory that doesn't lose its data even when the device is powered off. The content of the flash memory (i.e. the file system with all its data, including the operating system, programs and configuration data) is also called firmware. Replacing the firmware of the Netgear router means the content of the flash memory is erased i.e. the entire file system that was stored in it was replaced with an entirely new file system, which contains a completely new operating system, and a new set of programs and data files. Nothing from the old system will survive when the firmware is replaced. For the initial setup of the Netgear router the source code for the OpenWRT firmware is downloaded and cross-compiled it for the Broadcom CPU. Cross-compilation is a process where the compiler produces binary program files that are not meant to be executed on the local system, but rather on another platform (here the Netgear router with its Broadcom CPU). The Firmware in the router can be reflashed in one of the following ways

i. vendor supplied web interface
ii. TFTP (Trivial File Transfer Protocol)
iii. CFE (Common Firmware Environment)
iv. JTAG (Joint Test Action Group)
v. The OpenWrt commandline

There are two ways to replace the firmware in a NetgearWGT634U router

1. Building the firmware from the scratch and replacing the existing firmware.
This process involves creating a building environment for the router. Creating a proper environment requires building a tool chain. The tool chain is necessary to build an OpenWRT image and packages that can be installed on a Netgear device on a host system. The tool chain is nothing but the set of packages present in addition to the basic packages that are needed for the proper connection between the router and the computer. It is also needed for proper flashing of the image in the router.

2. Using precompiled packages to replace the firmware.
The precompiled packages can be found in the OpenWRT website packages. Both the processes require building a serial cable to access the router. Once the serial cable is built the existing Netgear firmware is replaced with the OpenWRT firmware. The OpenWRT router can then be connected to any mesh network after certain modifications in the OLSR protocol.

## 3. OpenWRT in QCS Mesh:

Any OpenWRT "vanilla" router can be made to join an existing QCS Mesh network. Later on configuration changes can be done to optimize the setup. Linux is found to be the best suited Operating system for working with OpenWRT because of its architecture. While working with Linux several packages may be required to make a router join a network. OLSR (Optimized Link State Routing) protocol is one of the important packages that need to be installed while working with the packages.

The building environment is first set before the router is allowed to join the mesh network. The website www.qcsmesh.com provides step by step instructions of how to create a 'build environment' that provides a platform for one to add additional packages to the already existing basic packages that OpenWRT provides. This freedom of customization allows one to use a device in many ways. The QCS Mesh routers in essence have their wireless 802.11 protocols used in the netgear routers replaced with OLSR protocol. This enables them to be a part of an ad-hoc mesh network where one of the routers acts as a gateway. This makes it possible to extend a physically constrained network connection, eg. an internet gateway, over to a larger area thus providing extensive connectivity.

### 3.1 OLSR in QCS Mesh:

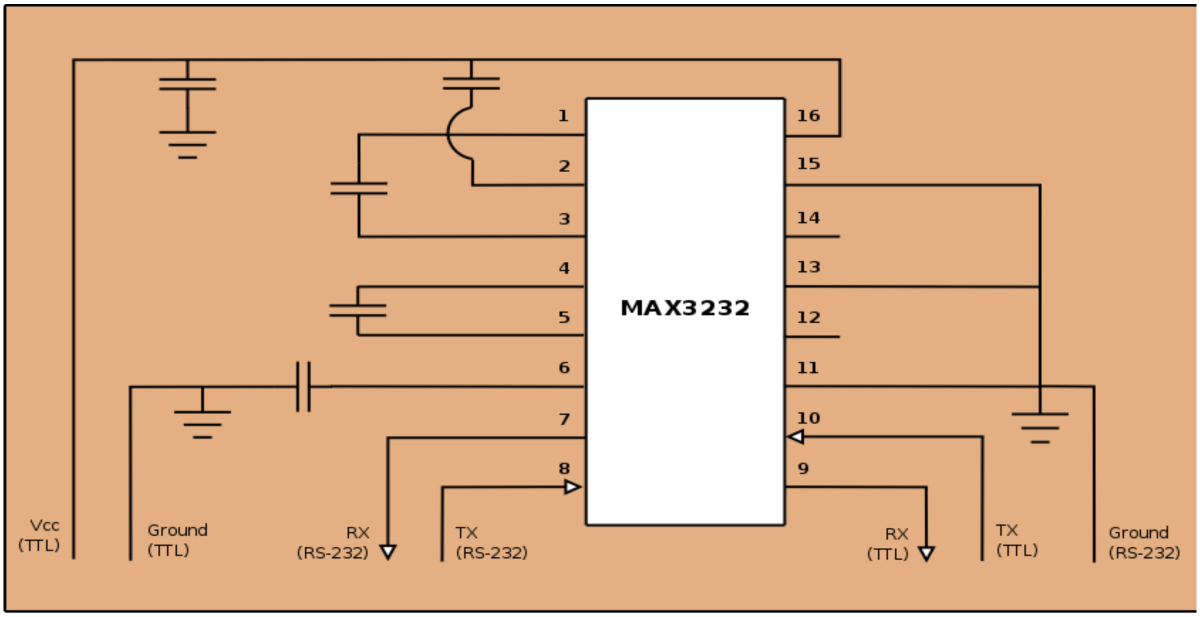
The Optimized Link State Routing Protocol (OLSR) is an IP routing protocol which is optimized for mobile ad-hoc networks but can also be used on other wireless ad-hoc networks. OLSR is a proactive link-state routing protocol which uses Hello and Topology Control (TC) messages to discover and then discriminate link state information throughout the mobile ad-hoc network. Individual nodes use this topology information to compute next hop destinations for all nodes in the network using shortest hop forwarding paths. The routing strategy of OLSR protocol is similar to traditional routing protocols. Each node broadcasts control packets periodically, exchanges nodes' information and discovers routes proactively and independently. The OLSR daemon provided by olsr.org is an implementation of the Optimized Link State Routing protocol. It provides for routing of mobile ad-hoc networks and is driven by a technique called multipoint relaying. Olsrd is a well structured and well coded implementation that is easy to maintain, expand and port to other platforms. Being a proactive protocol, routes to all destinations within the network are known and maintained before use. Having the routes available within the standard routing table can be useful for some systems and network applications as there is no route discovery delay associated with finding a new route. The routing overhead generated, while generally greater than that of a reactive protocol, does not increase with the number of routes being used.

### 4. Related Work:

The Netgear firmware in the router is replaced by the OpenWRT firmware that is obtained from the website www.openwrt.org. The Netgear router is connected to the computer by building a serial cable. The process is elucidated below.

### 4.1 Building a serial cable:

The first step in building a serial cable is determining the voltage level required by the device and choosing



Schematic Diagram of Serial Console

components accordingly. Some devices have standard RS232 +/- 12V serial ports, but in many OpenWRT-supported devices the serial ports operate at TTL voltage (sometimes 5V, but more often 3.3V) levels. In order for the serial console to work, the logic levels on the wires should match those expected by the device. It should be ensured that the Voltage supply of the pc and the router are connected. Similarly the pc and the router should be grounded together. The transmit data of the pc should be connected to the receive data pin of the router and the transmit data of the router should be connected to the receive data of the pc. The pc can be connected to the router using a prebuilt cable or a cell phone data cable or using ICs to convert to the necessary voltage levels.

The TTL voltage of the NetgearWGT634U is found to be 3.3 V. Hence a MAX 3232 is used to convert the RS232 voltage level to 3.3 V. The pc is connected to the MAX3232 through a USB to serial cable and the output from the MAX3232 is fed to the J7 serial pins in the router through a serial to serial cable.

### 4.2 Replacing the firmware with a precompiled OpenWRT package:

The process of replacing the firmware with a new one is called flashing the router. A TFTP (Trivial File Transfer Protocol) server and DHCP (Dynamic Host Configuration Protocol) server is run beforehand. **Dynamic Host Configuration Protocol** (**DHCP**) is a network application protocol used by devices (*DHCP clients*) to obtain configuration information for operation in an Internet Protocol network. This protocol reduces system administration workload, allowing devices to be added to the network with little or no manual intervention.

Flashing the router has the following steps

1. The serial console cable is attached to the WGT634U.
2. The Ethernet cable is attached to the WAN port of the router found next to the USB port.
3. A terminal emulation program like minicom is run on the computer. It is very important that the terminal emulation program should be chosen such that the it is compatible with the pc and the device.
4. A SSH (secure shell) access has to be established between the pc and the router for secure communication between them. **SSH** is a network protocol that allows data to be exchanged using a secure channel between two networked devices.
5. The router is then flashed using the flash command. The router after rebooting is found to have the new modified OpenWRT firmware.

## 5. Conclusion:

The Netgear router is connected to the computer by serial console which is built by using Integrated Chips and serial cables. The default Netgear firmware in Netgear WGT634U is replaced by the OpenWRT firmware after the connection is established between the router and the computer. The router with the modified firmware can now be connected to a mesh network by using the OLSR protocol.

## 6. References:

[1] Yan Wen, Guo Wein and Liu Jun, “An Implementation and Study of OLSR protocol in Linux OS”, 4th International Conference on Wireless Communications, Networking and Mobile Computing 2008, WiCom ’08, Oct 2008.

[2] Zheng Li, Nenghai Yu, Zili Deng, “NFA: An Algorithm to select MPRs in OLSR”, 4th International Conference on Wireless Communications, Networking and Mobile Computing 2008, WiComm ’08, Oct 2008.

[3]http://wiki.openwrt.org/OpenWrtDocs/Hardware/Netgear/WGT634U

[4] http://sarwiki.informatik.hu-berlin.de/Hacking_the_Netgear_wgt634u

[5] Dang Nguyen and Pascale Minet, “Analysis of MPR selection in the OLSR protocol”, 21st International Conference on Advanced Information Networking and Application Workshops (AINAW’07), 2007.

[6] Frank Y. Li, Lorenzo Vandoni, Giampietro Zicca, and Stefano Zanoli, “OLSR Mesh Network for Broadband Access: Enhancements, Implementation and Deployment”, 4th IEEE International Conference on Circuits and Systems for Communication, 2008. ICCSC 2008. May 2008.

[7] Mounir Frikha, and Manel Maamer, “Implementation and Simulation of OLSR protocol with QoS in Ad Hoc Networks”, Higher School of Communication of Tunis, Network Department.

[8]http://wiki.openwrt.org/OpenWrtDocs/Hardware/Netgear/WGT634U

[9] A. Leon-Garcia and I. Widjaja, Communication Networks: McGraw-Hill, 2004.

[10]http://www.qcsmesh.com/twiki/bin/view/QCSMesh/UploadingFirmware

[11]http://qcsmesh.com/twiki/bin/view/QCSMesh/QCSMeshConfig

[12] http://downloads.openwrt.org/

[13] Mesh computer algorithms for computational geometry – Russ Miller and Quentin F Stout.