**Related Literature**

**Routing Protocols**

Routing protocols dictates where and how the router distributes packets. There are a lot of different routing protocols currently available and each of them has its own design and serves different purposes (Cisco Networking Academy, 2014).  
The key considerations that raised the need for efficient routing when implementing mesh networks are the overhead of ID per hop jumped, maintenance of nodes, send/receive overhead, power consumption, and interference. It is also important to consider that table based protocols grows bigger as nodes increase and packet header grows bigger as more nodes are included when choosing the proper routing protocol for your network ().

**Current classifications of routing protocols that are relevant to the study**

**Dynamic Routing Protocols –** According to Cisco Networking Academy (2014), this “…allow routers to dynamically share information about remote networks and automatically add this information to their own routing tables”. Because of this sharing of information, the network automatically adapt with its topology. This also allows routers to discover new networks and repair broken ones.

**I.** External Gateway Protocols (EGP) **–** Used for routing systems that are handled by different organizations.

**II.** Interior Gateway Protocols (IGP) **–** Used for routing routers that are handled by a single organization.

**a.** Distance Vector Routing Protocols –Routers that use this routing protocol are not aware of the entire map of its network. The router only knows the distance and the vector to a device connected to the network. Vector is the direction of the next hop and distance is how many hop counts or bandwidth or some other metrics, it will take to reach a destination (Cisco Networking Academy, 2014). Most routing protocols that are under this classification enable its routers to send periodic updates to all devices in the network to maintain a relevant distance and vector values (Thomas, 2008).

**a.i.** Ad-hoc On-Demand Distance Vector Routing (AODV) **-** a Distance Vector routing protocol specifically designed for mobile ad-hoc networks. Nodes only search for a route when it needs to transmit/retransmit a message, hence on-demand. It does not need periodic advertisements and only uses connection when needed; this means that there is less traffic in the network allowing it to have a bandwidth that is significantly higher than other routing protocols (Perkins & Royer 2003).

**a.ii.** Babel – a Distance Vector routing protocol that uses periodic updates. To reduce the overhead caused by these updates, it “… allows to omit subnet prefixes when multiple addresses are sent in a single packet…” (Vinas et al., 2012). Babel is used in both wired networks and wireless mesh networks. It has a “…loop avoiding distance-vector routing protocol that is designed to be robust and efficient…” This loop avoiding property makes it ideal for networks that have unstable connections. However, this also causes performance degradation for huge environments with stable network. Babel can be configured to route networks using the shortest path or other metrics. It also uses a hybrid routing protocol that can be used by IPV4 or IPV6 networks (Chroboczek, 2011).

**b.** Link State Routing Protocols – Routers that use this routing protocol is aware and “has a complete view” of all the connections in its network (Cisco Networking Academy, 2014). It first completes a map of the entire network then calculates the best path to a destination using an algorithm. This process allows updates to converge faster at the expense of additional overhead due to the flooding it causes(Alberghetti, 2015).

**b.i.** Optimized Link State Routing Protocol (OLSR) – A revision of Link State routing protocol. OLSR is designed for mobile ad hoc networks wherein all devices connected to the network act as a node. One of its main differences from link state routing is that every node in OLSR sets a multipoint relays (MPR) (Clausen & Jacquet, 2003). Nodes will only receive transmissions from these MPR and allows them to control its traffic (Retrieved in August 17, 2016 from https://www.youtube.com/watch?v=3V19nPxpMp8). MPR are selected in such a way that the node that selected them will receive all transmissions in the network but with less duplicates. These duplicates contribute to the flooding that is experienced by nodes in link state routing.

**Broadband-HamnetTM –** “a high speed, self discovering, self configuring, fault tolerant, wireless computer network…” (Kinter, 2010) They are formerly called HSMM-MeshTM (High-Speed Multimedia), and their main motivation is giving communication during emergency situations given that their network can act as an ad hoc. They provide network that uses OLSR and is currently supported by different Linksys routers and Ubiquiti radios. They do, however, encourage users to test their work on other devices (Retrieved in August 17, 2017 from http://www.broadband-hamnet.org/images/hsmm\_docs/WRT54Shop.pdf).

HSMM-Pi is an open source tool that enables Raspberry Pi and BeagleBone to connect to other devices such as routers and radios with the mesh network provided by HSMM. It is tested in Raspbian OS, Debian with BeagleBone, and Ubuntu 12.04 in BeagleBoneBlack. For the hardware, it needs a WiFi adapter and an SD card to run. An HSMM-Pi node can be set to Mesh Gateway Mode that routes internet traffic from an Ethernet port or as an Internal Mode that routes traffic across the mesh network (Smith, 2016).

**Commotion –** an open source networking tool that provides mesh networks. Their objective is to create a tool that can be set up and used by anyone. Commotion can share internet access, applications, and files when one of the nodes has it. A lot of its properties are hardware and situation dependent but it is possible to connect thousands of nodes together. Commotion is supported in different routers, Linux and Mac computers, and rooted Android phones (Retrieved in August 18, 2016 from https://commotionwireless.net/).

**Debian –** created by Debian Project, people who aim to develop a free operating system. It currently uses the Linux or FreeBSD kernel. It contains over 43000 free packages that enable users different functionalities and a level of flexibility to their devices (Retrieved in August 18, 2016 from https://www.debian.org/).

**Raspberry Pi Operating Systems**

**Raspbian –** closely based on Debian that is optimized for Raspberry Pi. It is completed in 2012 but is still in active development. Raspbian was created by developers who are fans of Raspberry Pi and the Debian Project. A lot of documentations of Debian are applicable in Raspbian since the only differences that the two have are caused only by the hardware differences of a computer and a Raspberry Pi (Retrieved in August 18, 2016 from https://www.raspbian.org).

**Arch Linux –** an independently developed Linux distribution that prides itself for its lightweight characteristic. Things are kept simple and bare with users doing most of the job themselves. It does not even have a default Graphical User Interface after installation (Retrieved in August 18, 2016 from http://elinux.org/RPi\_Distributions). This can give a certain level of difficulty to beginners. It first came out as a computer operating system but is later modified to run in ARM-based devices (Retrieved in August 18, 2016 from https://wiki.archlinux.org/index.php/).

**References**

1. Kozierok C., 2005, The TCP/IP Guide, <http://www.tcpipguide.com/free/t_OverviewOfKeyRoutingProtocolConceptsArchitecturesP.htm>. Retrieved August 16, 2016.

2. Perkins & Royer, 2003, <https://www.cs.cornell.edu/people/egs/615/aodv.pdf>, Retrieved August 16, 2016.

3. Cisco Networking Academy, 2014 <http://www.ciscopress.com/articles/article.asp?p=2180210&seqNum=4>, Retrieved August 16, 2016.

4. Thomas J., 2008 <http://www.omnisecu.com/cisco-certified-network-associate-ccna/introduction-to-link-state-routing-protocols.php>, Retrieved August 17, 2016.

5. Alberghetti, D, 2015 <http://www.danscourses.com/CCNA-2/link-state-routing-protocols.html>, Retrieved August 17, 2016.

6. Clausen T, Jacquet P., 2003 <https://www.ietf.org/rfc/rfc3626.txt>, Retrieved August 17, 2016

7. Chroboczek J, April 2011, <https://tools.ietf.org/html/rfc6126>, Retrieved August 17,2016.

8. Vinas R. B., Terren D. R., Neumann A., Lopez E., September 12, 2012, Evaluation of Dynamic Routing Protocols on Realistic Wireless Topologies, <http://www.bmx6.net/attachments/129/master_thesis__evaluation.pdf>, Retrieved August 17, 2016.

9. Kinter J., January 2010, <http://www.broadband-hamnet.org/>. Retrieved August 17, 2016.

10. Smith C., May 2016, HSMM-Pi, <https://github.com/urlgrey/hsmm-pi>, Retrieved August 18, 2016.

11. Open Technology Institute, <https://commotionwireless.net/contact/>, Retrieved August 18, 2016

12. Debian, https://www.debian.org/ , Retrieved August 18, 2016

13. Raspbian, <https://www.raspbian.org>. Retrieved August 18, 2016

14. Arch Linux, https://wiki.archlinux.org/index.php/ , Retrieved August 18, 2016

15. RPi Distributions, http://elinux.org/RPi\_Distributions , Retrieved August 18, 2016