

Heterogeneous Wireless Mesh Network Technology Evaluation for Space Proximity and Surface Applications

Michael A. DeCristofaro,¹ Chatwin A. Lansdowne,² and Adam M. Schlesinger³
NASA Johnson Space Center, Houston, Tx, 77058

NASA has identified standardized wireless mesh networking as a key technology for future human and robotic space exploration. Wireless mesh networks enable rapid deployment, provide coverage in undeveloped regions. Mesh networks are also self-healing, resilient, and extensible, qualities not found in traditional infrastructure-based networks. Mesh networks can offer lower size, weight, and power (SWaP) than overlapped infrastructure-per-application. To better understand the maturity, characteristics and capability of the technology, we developed an 802.11 mesh network consisting of a combination of heterogeneous commercial off-the-shelf devices and open-source firmware and software packages. Various streaming applications were operated over the mesh network, including voice and video, and performance measurements were made under different operating scenarios. During the testing several issues with the currently implemented mesh network technology were identified and outlined for future work.

I. Introduction

WIRELESS mobile ad-hoc networking technology has evolved during the last dozen years from research to standards and implementations. Applying mesh architecture to phase out the assortment of dedicated point-to-point links used by NASA today is considered a necessary shift in strategy. The technique will enable NASA to explore new destinations with reduced preparation and with reduced precursor infrastructure.

We therefore conducted a preliminary investigation of the maturity of the technology by constructing and evaluating a prototype. From this experience we will draw some conclusions and identify important areas for additional investigation, research, and development.

Heterogeneous wireless mesh network technology enables a cooperative approach to cognitively managed communication architecture. Such a framework provides more versatility for low-cost exploration missions, agile re-configurability that is lacking in today's architecture of bespoke, configuration-managed point-to-point per-application links. NASA Johnson Space Center is using Wi-Fi to begin exploring low infrastructure uses of one such technology, Mobile Ad-hoc NETWORKS (MANETs), for application in current and future space exploration. Heterogeneous meshes differ conceptually from the proprietary wireless meshes that NASA has evaluated in the past. The proprietary mesh forms a closed communication backbone to which heterogeneous systems connect in infrastructure mode. With an open architecture, the clients directly form their own ad hoc infrastructure by cooperating. This approach enables agile, minimally managed reuse of fielded assets without pre-deployment of a communication infrastructure.

¹ Co-op Student, Electronic Systems Test Laboratory (ESTL), Avionic Systems Division (ASD), 2101 NASA Parkway/EV611, Houston, TX.

² Test Director, Electronic Systems Test Laboratory (ESTL), Avionic Systems Division (ASD), 2101 NASA Parkway/EV611, Houston, TX.

³ Communication Systems Engineer, Wireless and Communication Systems Branch in the Avionic Systems Division at NASA-JSC, 2101 NASA Parkway/EV611, Houston, TX.

Mesh networking could usefully be understood as cognitive bridging. Bridges can be used to achieve essentially the same configurations, including redundant connections automatically selected among using a Spanning Tree Algorithm. Bridges however are not cognitive, and all of the links must be pre-declared; when nodes are added or moved, the bridge definition must be managed. Mesh points automatically sense their environment, establish a network, compute optimized routes, and manage themselves, and they require considerably more computing power to support their greater intellect.¹

In a mesh network, each node can automatically discover its neighbors and acts as a relay. Automatic neighbor discovery allows network traffic to be dynamically routed around any failures in node communication. We reviewed considerable literature and explored various options for constructing and deploying such a mesh network. There is a variety of available open firmware for various embedded platforms such as a router with DD-WRT² or OpenWRT³. Following the Katrina and Arab Spring events⁴, packages such as Quick Mesh Project (qMp) and Byzantium offer simplified installation to provide a quick automated deployment of a mobile mesh network. The capabilities and differences of Proactive, Reactive and Hybrid mesh routing protocols, including HWMP, B.A.T.M.A.N and B.A.T.M.A.N-Advanced, OLSR, and BABEL will be discussed, along with the rationale behind the selection of B.A.T.M.A.N-adv and HWMP as the primary routing protocols for investigation.

Over the past dozen years, the potential of wireless mesh networks has evolved from a vision to a reality. The community supporting mesh networks is constantly growing and adapting concurrently with the IEEE standards development and additional research. Enthusiast communities, dominantly European, organize on-line, and evaluate their work through annual events such as BattleMesh⁵. In this paper, we seek to identify a path to deployment of today's mesh technology into today's space exploration program.

II. Technology Survey

As a community of researchers and enthusiasts has explored the technology over the years, a number of protocols and implementations have been developed; some have faded into disuse, some continue to have active interested communities, some are seasoned, and some are promising. We provide references to comparisons we found helpful, while providing here only an overview of routing protocols and packaged implementations.

A. Mesh Protocols

The protocols in general do not require a specific routing strategy be implemented at each node. However, each protocol does define what message formats are used and what information is available for computing route metrics. Protocols are classed as proactive, reactive, or hybrid, depending on whether routes are established on-demand or monitored and maintained.

1. OLSR

Optimized Link State Routing (OLSR) gained tremendous early momentum. OLSR is one of the oldest mesh routing protocols in existence. It is defined by the Internet Engineering Task Force (IETF) Request for Comment (RFC) 3626, and was once the leading candidate routing protocol for IEEE 802.11s.⁶ Performance studies show that newer protocols are more scalable and agile with lower overhead.⁷ Versions now exist with link quality sensing⁸. OLSR is a proactive layer 3 protocol, which offers portability to more systems and enables the protocol to span link layers. OLSR has one of the bigger mesh networking communities to help promote its development and provide support.

2. *BATMAN and BATMAN-Advanced*

BATMAN once attracted widespread support and is today compiled into most versions of the Linux kernel. It is defined in an expired IETF draft working document⁹. BATMAN is a proactive layer 3 protocol, routing on IP addresses, and BATMAN-Advanced is a layer 2 protocol, self-assembling using MAC addresses. Distributed knowledge of topology makes BATMAN efficient and scalable, but the routing cost metric is simply the number of hops, weighted by a hop penalty advertised by the nodes. BATMAN-Advanced has an active open source community that promotes its development and provides support.

3. *HWMP*

Hybrid Wireless Mesh Protocol (HWMP) is the standard in IEEE 802.11-2012 (which supersedes 802.11s). Today it is compiled into most versions of the Linux kernel. Curiously, we did not find a satisfactory performance comparison explaining the selection of HWMP over competitors for 802.11s. Secure HWMP (SHWMP) and Privacy Aware SHWMP (PA-SHWMP) propose a method for nodes to authenticate each other to prevent outside mischief, but this appears to be a research area and we did not find implementations¹⁰. HWMP also has an open source implementation open802.11s that provides a relatively weak supporting community for development and use.

4. *Babel*

Babel has been shown to assemble and heal twice as fast as BATMAN, and handle more traffic through more efficient route selection⁷. Babel is defined by an experimental IETF RFC 6126.¹¹ Babel uses many of the same route cost metrics as HWMP, but does not require a specific routing strategy and there are several that could be implemented; it is considered proactive but has reactive features. Babel implementations are available for Linux and OpenWRT but we did not find Babel built-in. We did not find a scholarly direct comparison of Babel and HWMP performance. Babel is one of the newest mesh routing protocols. From our observations we saw that Babel had one of the smallest support communities driving development.

B. Mesh Implementations

5. *DD-WRT*

DD-WRT is easily the most popular open-source router firmware today, serving a community that chooses its powerful simple consistent interfaces to make their projects more manageable. It coexists symbiotically with a list of about 700 reprogrammable router platform models produced by 97 hardware brands from 3Com to ZCom.¹² We therefore spent a little time examining the ad-hoc features of DD-WRT, but abandoned the attempt when we realized that DD-WRT presently provides minimal support for mesh routing protocols.

6. *OpenWRT*

OpenWRT is a Linux platform that underpins DD-WRT and many other router firmware packages. Although the user interface is relatively undeveloped, command line control allows protocol packages to be installed and configured by competent users.

7. *Microsoft Windows*

Although some laptops on the International Space Station switched from Microsoft Windows to Debian 6 Linux earlier this year¹³, Windows 7 remains a platform of interest for the Station Support Computer (SSC) crew laptops. Microsoft explored mesh networking early on, creating and evaluating a “Mesh Connectivity Layer” (MCL) in 2004¹⁴. This abandoned software used a refined version of DSR, which considered link quality when selecting routes. Perhaps more

importantly, it packaged the protocol as a network driver, which manifests to the user as a virtual network adapter while transparently utilizing one or several physical adapters.

8. *Android*

Android and iPhone platforms are of continuing interest because of their dense, power-efficient integration of sensors, processing, and communications. In the SPHERES robots for example, a Nexus S handset with Velcro provided intelligence and vision. PhoneSat¹⁵ is an example of Android controlling an independent satellite. Google itself appears to have shown no interest in mesh networking on Android, but several communities have. The BatDROID app supplies a layer 3 protocol. Another development to watch is Smart Phone Ad-Hoc Networks (SPAN), which has initially implemented OLSR-D on layer 3 and generated a custom kernel for the Samsung Galaxy S III.¹⁶

9. *Arduino*

Some initial work was performed to port the OpenWRT operating system onto the Yun platform, enabling the Yun to participate in a mesh network.¹⁷

10. *Raspberry Pi*

Airmesh succeeded in integrating the BATMAN-Adv layer 2 mesh protocol into Raspbian¹⁸. OpenWRT has been ported to the Raspberry Pi platform¹⁹, enabling it to serve as a mesh router. And Layer 3 protocols such as Babel have been run directly on the Raspberry Pi²⁰.

11. *Mesh Packages*

Enthusiast communities have produced a number of packages to serve the needs of community networks. QuickMesh Project, AirMesh, Byzantium, Project Meshnet, Commotion, and others offer package solutions usually supporting one or two popular protocols, as in Table 1. Some open source software exists for managing community networks; the Wireless Nodes Database (WiND) is being used around the world.²¹

Table 1. Packaged Open Source Mesh Protocols

| <i>project</i> | <i>mesh protocol</i> | <i>addressing</i> | <i>platform(s)</i> | <i>URL</i> |
|-------------------|----------------------|-------------------|----------------------------|---|
| QuickMesh | BMX, OLSR | IPv6 auto | OpenWRT | http://qmp.cat/ |
| airmesh | BATMAN-Adv | IPv6 CJDNS | Linux | http://www.netlore.co.uk/airmesh/ |
| Byzantium | OLSR | IPv6 ZeroConf | Linux | http://project-byzantium.org/ |
| Commotion Project | OLSR | IPv4 custom | OpenWRT Linux, Android, | https://commotionwireless.net/ |
| Meshnet | CJDNS | IPv6 CJDNS | OSX, more | https://projectmeshnet.org/ |
| Freifunk | choose | plug-in | OpenWRT | http://freifunk.net/ |

III. Comparative Evaluation of BATMAN-Advanced and HWMP Meshes

Since BATMAN-Advanced and HWMP were available for many of the target platforms, we began exploring questions of ease-of-use, performance, mobility, and security.

A. Test Articles

Since HWMP is included in the Linux Kernel by default, it need only be configured. So, the first step was installing BATMAN-Advanced on a set of eight Wi-Fi routers and six laptop computers in a mesh configuration. Physical layer protocol (b/g/n) was set to auto-select. Channel selection could be set initially on auto-select. Some applications used to evaluate video and audio streaming performance as the topology of the mesh readapted were VLC media play

and Ekiga open source VoIP SoftPhone. IPv4 with fixed addressing was used for the mesh configuration. Throughput and Packet Loss were measured using the ‘iperf’ network testing tool. Security was added using WEP and a shared single key. Mesh reconvergence times were measured using ping.

The mesh was heterogeneous in the sense that both routers with OpenWRT and laptops with Linux could communicate on the same mesh. As long as all of the devices were running the same mesh routing protocols at the same time there were no compatibility issues. For BATMAN-Adv, the routers and laptops were using different routing protocol software versions but backwards compatibility is included automatically so this did not pose any problems.

12. OpenWRT

HWMP is built into the Linux kernel. To install BATMAN-Adv, the `kmod-batman-adv` package was added from the standard available OpenWRT packages using the ‘opkg’ package manager.

13. Linux

To evaluate HWMP we needed only to configure each device as a mesh node. This was accomplished using the ‘iw’ and ‘ifconfig’ Linux commands. In the distribution of Linux that we were using, the BATMAN-Adv Batctl control program was easily downloadable from the package manager. The kernel module on the other hand had to be compiled from scratch. This allowed us to choose the specific version of the BATMAN-Adv kernel module that we wanted to use. In this case we specifically chose a different version than what was included in the OpenWRT package to show additional compatible interoperability.

B. Test Techniques

A variety of indoor and outdoor techniques were used to test the system. In the indoor scenario, video and audio streaming were evaluated using VLC and Ekiga open source software packages and techniques. In this environment, it was also possible to perform instantaneous node failures and observe rerouting time. The basics of the internal test involved four nodes: two upstairs in our offices and two downstairs in our lab. This was a four node heterogeneous mesh using 1 laptop and 1 router as nodes per room. Initially the laptops were used as the shorter hop to the routers and the routers were used to communicate the longer distance through the floor. This setup was used to test multiple hop mesh bandwidth in a linear manner. After testing the bandwidth of this mesh, additional redundant nodes were added and then other nodes were removed from the network to test mesh reconvergence. Router nodes were unplugged to see how long it took for the laptop to connect to the other router. This test was repeated multiple times to gather a significant data set from which to get an average reconvergence time.

Additional structured and controlled testing was performed outdoors on NASA’s 750m antenna range. The nodes were set up at an even spacing of about 150 meters. Throughput was evaluated with increasing numbers of hops. To observe the route switching behavior in a structured way, a mobile device was driven down the line, providing information on when the nodes decided to switch, where they switched to, and how long the route switching took. Bandwidth was measured using a UDP iperf connection.

C. Test Results

14. Audio, Video, Capacity, and Mobility

The Ekiga VoIP software did not accommodate mobile nodes; if the mesh protocol changed the route between the two communicating nodes then Ekiga would drop the call. This problem

did not have a simple solution, and it was determined to be a flaw in the way Ekiga adapts to latency changes. During the indoor testing the focus of the investigation was audio and video applications using the mesh. Using VLC the quality of the video was very low. We were restricted to a stream with a resolution of 500x500 pixels to maintain a high enough frame rate that the video was usable. The audio quality on the other hand was very high. Higher resolution video streams were less tolerant of dropped frames and dropped packets causing extremely low frame rates.

15. Throughput

The bandwidth performance across multiple hops can be viewed in the graphs below. Throughput dropped as hops were added, more so for BATMAN-Adv than for HWMP. The throughput in the indoor test was limited by going through the floor. Table 2 shows that the bottleneck of 9.5 Mbps coupling between floors was not further reduced by additional hops.

In the outdoor test we knew that bandwidth would be limited by distance due to fixed transmit power. The initial connection was between a laptop and proximate router, then the routers were spaced uniformly at 150m. iperf was used to successively load the mesh between increasingly distant successive nodes. Analyzing the data in Figure 1 shows that BATMAN-Adv had a 45% bandwidth decrease per hop at the performance rate of 1% packet loss. The HWMP test data shows a 34% bandwidth decrease per hop at the performance point of 1% packet loss per hop. This effect was independent of antenna geometry and node placement, and is a property of the 802.11-2012 protocol or the OpenWRT implementation of 802.11s.

Table 2. Indoor Bandwidth Test

| Number of Hops | BATMAN-Adv Bandwidth | HWMP Bandwidth |
|-------------------|----------------------|----------------|
| 1 | 52.8 Mbps | 84.2 Mbps |
| 1 (Through Floor) | 9.91 Mbps | 9.34 Mbps |
| 2 (Through Floor) | 9.35 Mbps | 9.55 Mbps |

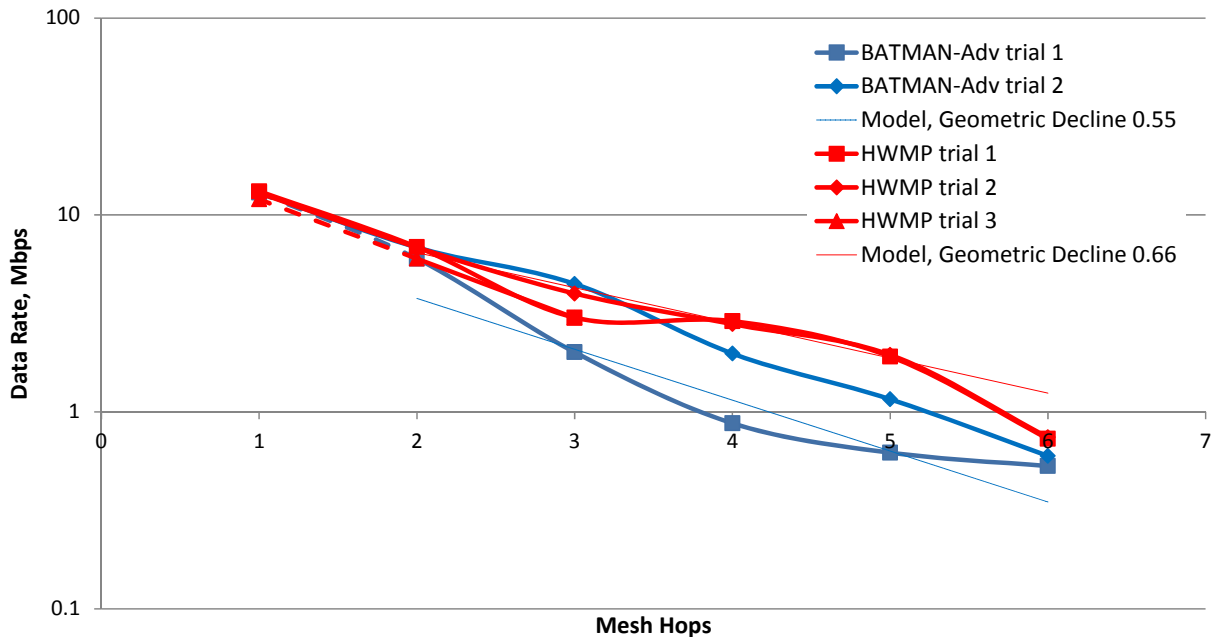


Figure 1. Mesh Bandwidth vs. Separation, 150m Spacing, Single Packet Stream

16. Route Selection and Switching

In the indoor test when a node had an instantaneous failure from loss of power, BATMAN-Adv reconfigured in an average of 0.743 seconds and HWMP in 0.443 seconds; both of these times were within an acceptable range.

However in the field, link quality fades gradually as the mobile node moved through the mesh. BATMAN-Adv resisted switching to a new route when link quality degraded due to low signal strength. BATMAN-Adv would quickly change its routing structure when selecting a route that had fewer hops to its destination node. The problem specifically occurred when updating the route to add an extra hop would have caused the throughput of the mesh to increase, but BATMAN-Adv prefers to take the route with the shortest number of hops even if it means that bandwidth to the destination node and the total capacity of the channel are less because of it. BATMAN-Adv had a 3.2s average handoff while moving in a direction that would reduce the number of hops to the destination node. The problem was the 75s average handoff while moving in a direction that would increase the number of hops, a long-tail distribution with some observed waits of over four minutes.

Table 5. Indoor Reconvergence Test

| Batman-Adv | HWMP |
|-------------------|----------------|
| Low: 120ms | Low: 140ms |
| Average: 743ms | Average: 443ms |
| High: 2120ms | High: 2660ms |

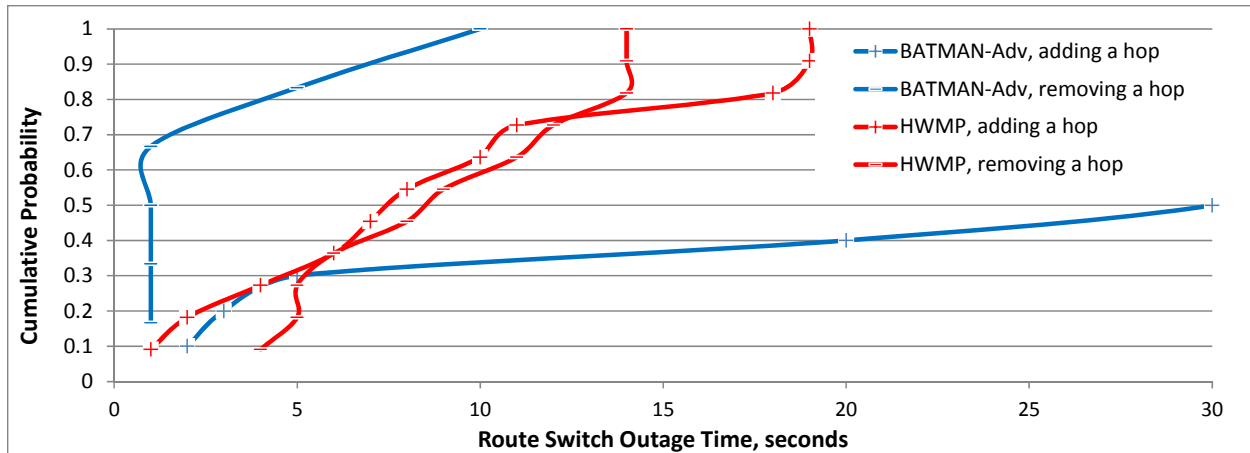


Figure 2. Route Switching Outage Time Using a Mobile Wi-Fi Node

HWMP had an average of 9.3 seconds while moving in a direction that reduces the number of hops to the destination and had 9.5s for the average of moving away from the destination node. The difference between these two tests is postulated to be because of the “radio-aware” messaging and route selection in HWMP. As well BATMAN-Adv is a Proactive routing protocol and HWMP uses a reactive protocol in this test because there are no gateway nodes.

17. Channel Selection

Most of the tests used a fixed channel assignment after manually assessing the RF environment. Additional testing may include scenarios with the nodes configured to automatically select a channel, as the RF environment can vary across a spatially distributed mesh, or vary with time. For example, can a node experiencing interference request a different channel? If the network establishes from two end-points on two frequencies, can a middle node force them to join or will they remain isolated? Basic mesh setup specifies a channel or frequency on which the mesh SSID resides. The mesh protocol tries to connect to the SSID on that channel, but if it cannot connect, it looks for the mesh on any available channels. Thus, a frequency agile mesh may be feasible.

18. Security

An evaluation of Wi-Fi AES and TKIP encryption found that clients or nodes could connect to the mesh without sharing the secret key, although they were configured to use encryption. The only wireless security standard that is implemented by OpenWRT mesh networking at the time of the test is WEP. WEP encryption is however a fundamentally broken protocol²². WPA could not be configured to support the mesh network, nodes would not connect to each other even if they had the same shared key.

19. Dynamic IP Address Assignment

NASA’s deployments are presently of a scale that fixed Internet Protocol address assignments can be managed. As cooperation with partner assets is a requirement of affordable space exploration, the address space must be partitioned among partners, or it must be managed by protocol. However, Table 1 shows that the fundamental problem of dynamic unique address assignment remains unsettled. This is in part because IP addresses reflect the structure of the network, and when a mobile node moves to a different part of the backbone its address should change by design.

III. Wi-Fi vs. WiMax

WiMax meshes by now have some maturity, as evidenced by the TRITON maritime mesh in Singapore Harbor.²³ And WiMax offers some performance advantages like much greater range, lower latency, better security, and faster mobile handoffs, although the hardware consumes more power.²⁴ Doppler is also a limitation on mobility.²⁵ However, we found that WiMax hardware today has greatly declined in availability. Wi-Fi hardware for demonstrations is plentiful. Rugged Wi-Fi hardware tends to be available only as part of a proprietary mesh or non-mesh solution and customers will need to require rugged hardware platforms that can run open software protocol implementations.

IV. Conclusion

This research demonstrated that given the current state of today's technology, simple mesh networks can be implemented. Although many protocols exist, none seem to be significantly further along in their development than the others. For the purpose of this study, BATMAN-Adv and HWMP were chosen because (1) they were the most convenient and easiest to set up and (2) they represented very different protocols.

Our results showed that COTS technology was able to transfer 0.5 Mbps of data through six hops over a distance of 750m with good data integrity. Distance and throughput could have been significantly improved by physical layer changes including directionality and transmit power, which add size, weight, complexity, and power consumption. However, both protocols had significant flaws. A technique that could be evaluated for reducing the bandwidth loss per hop is changing from Carrier Sense Multiple Access (CSMA) collision avoidance to Grant to Send (GTS).²⁶ It was observed that a wireless mesh works best when using a Radio Aware (RA) protocol—HWMP is radio aware and BATMAN is not. The difference is in the information contained in the message formats defined by the protocol, and simply put HWMP enables routes to be calculated using a metric that includes the data rate and packet loss rate, while BATMAN can only consider the number of hops. Neither protocol as implemented had sufficiently responsive route switching times to support mobile real-time applications. The 802.11-2012 ("802.11s") standard, HWMP, will be selected for follow-up evaluation, within an 802.11 mesh.

The significant current issues with the technology include mesh mobility, security, QoS, DTN, autonomous configuration, gateway congestion, and the number of competing protocols available. We found that the mesh must be treated as an unsecure network, and security must be provided by higher-layer protocols including certificate security, tunnels and bridges, secure shell (SSH), secure HTTP (HTTPS), disruption tolerant networking (DTN) bundle security protocol (BSP), or other mechanisms like PA-SHWMP which all remain to be explored.

The most important difference between mesh architecture and point-to-point architecture is that the links NASA uses today by design do not interact, while mesh architecture by design employs an interdependent network. In a spaceflight environment, prioritizing which types of data streams are transferred is critical. With the current implementations, there is no way to prioritize when bandwidth is limited. This raises the concern that critical applications will be unreliable. Spacesuits for example could benefit from the extended redundant coverage area provided by a mesh, but NASA presently does not trust that suit audio will always have adequate bandwidth when other applications like video share the link. Quality of Service mechanisms are intended to address this issue. We have not yet investigated QoS mechanisms in our mesh network, as this will require representative application traffic. It will also be necessary to ruggedize applications like audio and video for automatically recovered hands-free operation.

These will be significant next steps. Reconfiguration of the mesh needs to perform better and meshes should prioritize increasing bandwidth over fewer hops.

We expect that integrating mesh networking protocols with Disruption Tolerant Networking (DTN)²⁷ will enable reliable opportunistic communication that does not depend on dense infrastructure to provide coverage area. With a highly mobile network, not every node will be available to the mesh at all times. DTN will allow the nodes to store and then forward data when the destination node does become available. NASA is currently researching DTN for use in existing networks and in future mesh networks.

The need to focus on a single standard must be addressed in order to move this technology forward for space exploration systems. It is important for all parties to work collaboratively on high-performance reliable implementations of mesh networking technology to develop specifiable stable protocols that can support various mission scenarios.

In conclusion, we believe that wireless mesh networking can be adapted to support many non-critical and eventually critical space applications. The focus of ongoing research should aim to develop more robust open source applications that can be used to further advance human space exploration, and improve network performance. Meanwhile, the technology can be phased in by directing that Wi-Fi procurements be mesh capable or, preferably, the software is customer maintainable.

Acknowledgments

The authors would like to thank Brian De Leon Santiago for continuing this work and independently confirming our performance measurements. Brian additionally demonstrated an integration of 802.11s with DTN Bundle Protocol.

References

- ¹ Connect802 Corporation, "An Overview of Wireless Mesh Routing and Wireless Bridging Architecture," URL: <http://www.connect802.com/meshcompare.htm> [cited 4 February, 2014]
- ² URL: <http://www.dd-wrt.com> [cited 4 February, 2014]
- ³ URL: <https://openwrt.org/> [cited 4 February, 2014]
- ⁴ Ordonez, S., "Interview with Bryce Lynch from Project Byzantium on Mesh Networks," Wednesday, 31 July 2013 13:22, URL: <https://openitp.org/mesh-networks/interview-with-bryce-lynch-from-project-byzantium-on-mesh-networks.html> [cited 4 February, 2014]
- ⁵ URL: <http://battlemesh.org/> [cited 4 February, 2014]
- ⁶ Ghannay, S., et al., "Comparison of Proposed Path Selection Protocols for IEEE 802.11s WLAN Mesh Networks," *Wireless and Mobile Networking*, IFIP International Federation for Information Processing Volume 284, 2008, ISBN 978-0-387-84839-6, pp 17-28
- ⁷ M. Abolhasan, B. Hagelstein, J. C.-P. Wang, "Real-world Performance of Current Proactive Multi-hop Mesh Protocols," IEEE Proceedings of the 15th Asia-Pacific Conference on Communications (APCC 2009)-011
- ⁸ A. Capone, M. Cesana, S. Napoli and A. Pollastro, "MobiMESH: a Complete Solution for Wireless Mesh Networking," IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, vol. 0, pp. 1-3, 2007
- ⁹ A. Neumann et al., Better Approach To Mobile Ad-hoc Networking (B.A.T.M.A.N.) draft-wunderlich-openmesh-manet-routing-00, URL: <http://tools.ietf.org/html/draft-wunderlich-openmesh-manet-routing-00> [cited 4 February, 2014]
- ¹⁰ Lin, Hui et al., "PA-SHWMP: a privacy-aware secure hybrid wireless mesh protocol for IEEE 802.11s wireless mesh networks", EURASIP Journal on Wireless Communications and Networking 2012, URL: <http://jwcn.erasipjournals.com/content/2012/1/69> [cited 4 February, 2014]
- ¹¹ J. Chroboczek, "The Babel Routing Protocol", IETF RFC 6126, URL: <http://tools.ietf.org/html/rfc6126> [cited 4 February, 2014]
- ¹² DD-WRT "Supported Devices" wiki, URL: http://www.dd-wrt.com/wiki/index.php/Supported_Devices#Supported_Devices [cited 4 February, 2014].

- ¹³ Anthony, S., "International Space Station Switches from Windows to Linux, for Improved Reliability," ExtremeTech, May 9, 2013 9:21am
- ¹⁴ URL: <http://research.microsoft.com/en-us/projects/mesh/#software> [cited 4 February, 2014]
- ¹⁵ NASA Successfully Launches Three Smartphone Satellites, 22 April 2013, Release 13-107, URL: http://www.nasa.gov/home/hqnews/2013/apr/HO_13-107_Phonesat.html [cited 4 February, 2014]
- ¹⁶ The SPAN Project, URL: <https://github.com/ProjectSPAN> and associated forum URL: <https://groups.google.com/forum/#!forum/spandev> [cited 4 February, 2014]
- ¹⁷ Sexton, I., "Teensy/Arduino & OpenWrt," URL: http://www.cse.dmu.ac.uk/~sexton/ENGD2003/openwrt/arduino_teensy_openwrt.html [cited 4 February, 2014]
- ¹⁸ URL: <http://www.netlore.co.uk/airmesh/?page=about> [cited 4 February, 2014]
- ¹⁹ URL: <https://sites.google.com/site/variousopenwrt/rpi-patch> [cited 4 February, 2014]
- ²⁰ E. Erfanian, "Mobile Mesh Networks with the Raspberry Pi – Part 1", 11 September, 2012 URL: <http://www.ericerfanian.com/mobile-mesh-networks-with-the-raspberry-pi-part-1/> [cited 4 February, 2014]
- ²¹ Wireless Nodes Database wiki, URL: <http://wind.cube.gr/> [cited 4 February, 2014]
- ²² Arbaugh, W. A., "Your 802.11 Wireless Network Has No Clothes," *IEEE Wireless Communications* vol 9, issue 6, December 2002
- ²³ Pathmasuntharam, J. S. *et al.*, "TRITON: High Speed Maritime Mesh Networks", *The 19th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'08)*, Cannes, France, September, 2008, pp. 1-5
- ²⁴ Kabir, A. F. *et al.*, "WiMAX or Wi-Fi: The Best Suited Candidate Technology for Building Wireless Access Infrastructure," arXiv preprint arXiv:1208.3769 (2012).
- ²⁵ Zhao, Z., "Wi-Fi in High-Speed Transport Communications," *Intelligent Transport Systems Telecommunications (ITST)*, October 2009.
- ²⁶ Choi, J. *et al.*, "Inverting Wireless Collision Avoidance," Computer Systems Laboratory, Stanford University, Technical Report SING-09-00, 2009, URL: <http://sing.stanford.edu/pubs/sing-09-00.pdf> [cited 4 February, 2014]
- ²⁷ Khabbaz, M., Assi, C. and Fawaz, W., "Disruption-Tolerant Networking: A Comprehensive Survey on Recent Developments and Persisting Challenges," *IEEE Communications Surveys & Tutorials*, Vol. 14, No. 2, Second Quarter 2012

Heterogeneous Wireless Mesh Network Technology Evaluation for Space Proximity and Surface Applications

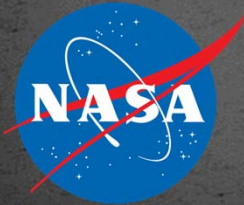
Michael DeCristofaro

The Pennsylvania State University

Chatwin Lansdowne

Adam Schlesinger

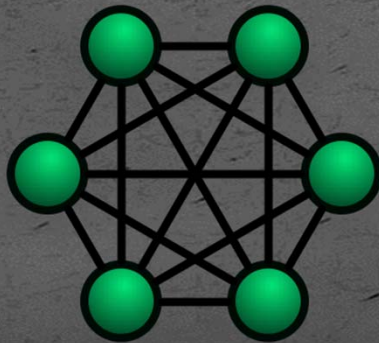
NASA Johnson Space Center





Wireless Mesh Networking

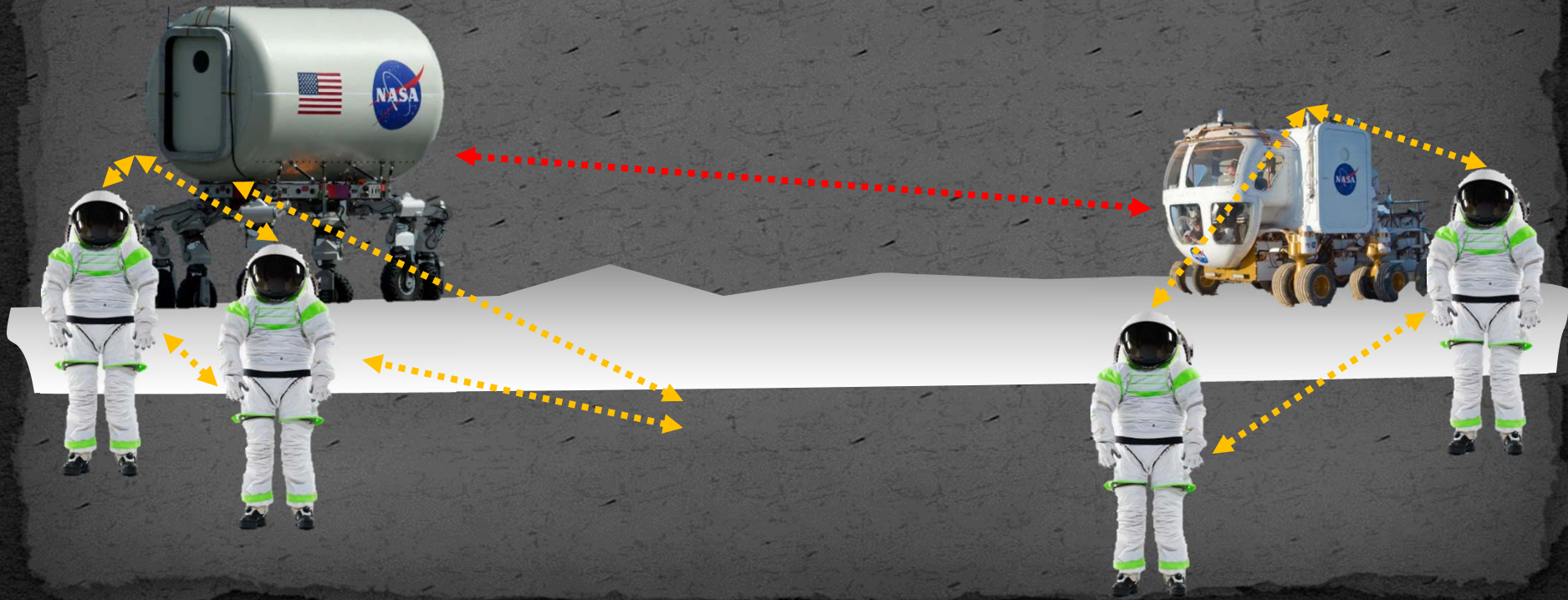
- Decentralized networking
- Does not rely on existing infrastructure
- Relies on nodes to route traffic to other nodes
- Self configuring / self healing routes



Heterogeneous

- Specifiable open standards
- Multiple vendors, implementations, or sources

Cognitive, Cooperative Communication





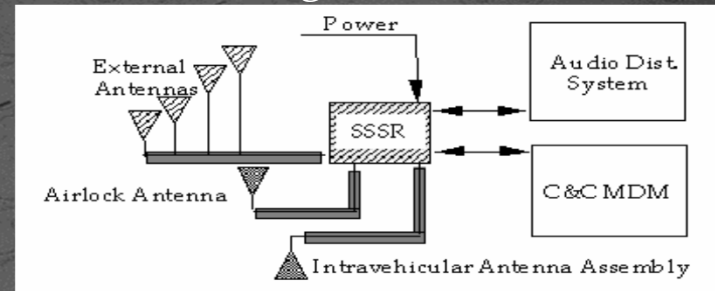
Why? Space Suits:

Add an Application

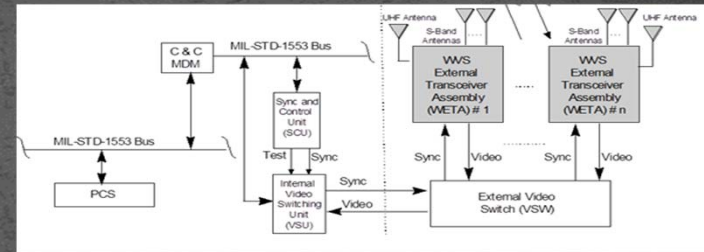
Add a Radio

Add Coverage Infrastructure

Audio



Video



More

Telemetry





Why? Visiting Vehicles:

Add A Visitor

Add a Radio

Add Coverage Infrastructure

CUCU



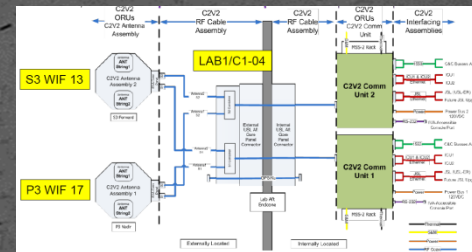
SpaceX Dragon



C2V2



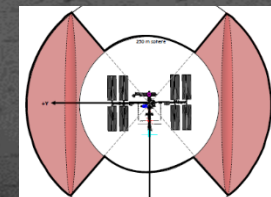
Cygnus



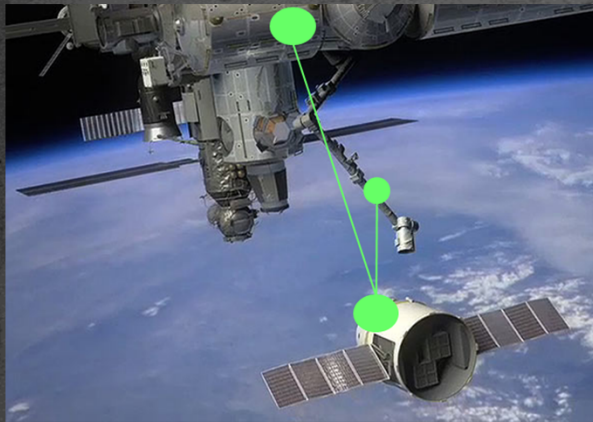
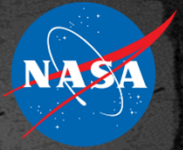
C3I



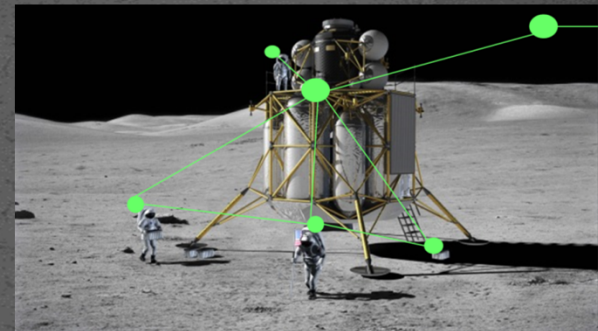
Orion



Mesh Networking Applications?



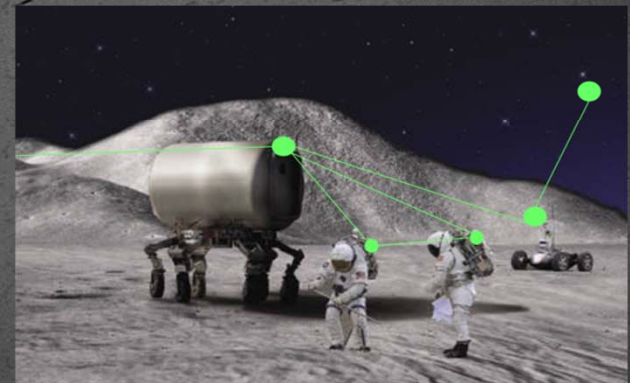
Docking



Landing

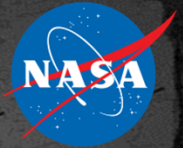


Servicing



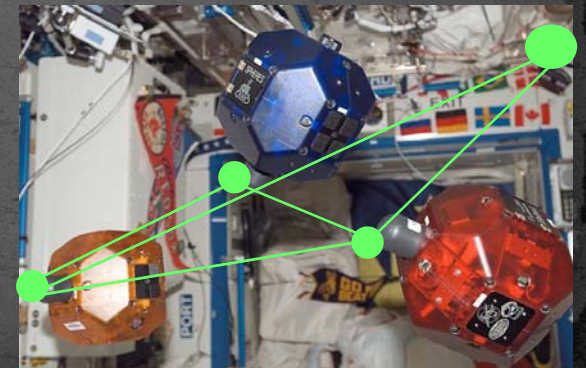
Exploring

Mesh Networking Applications?



Laptops, Personal Devices,
Intercom, Payloads,
Gateways

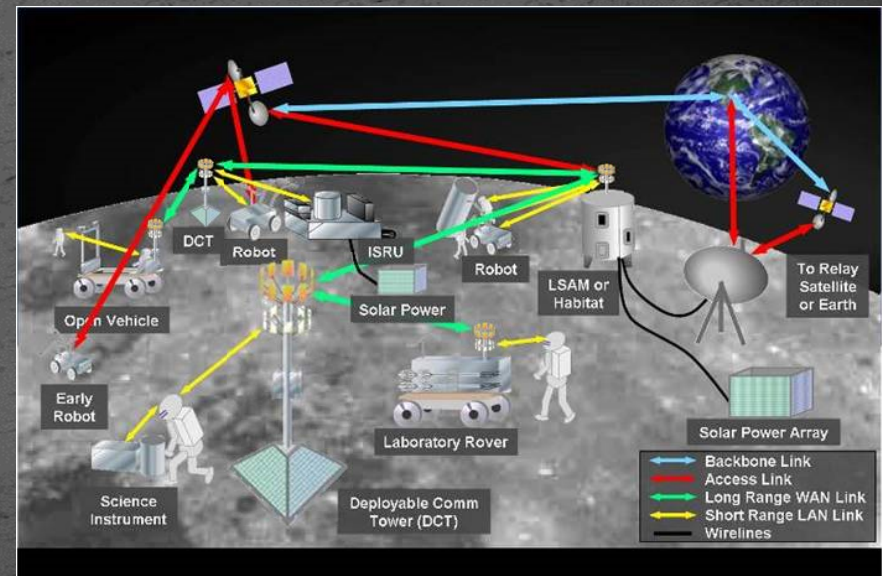
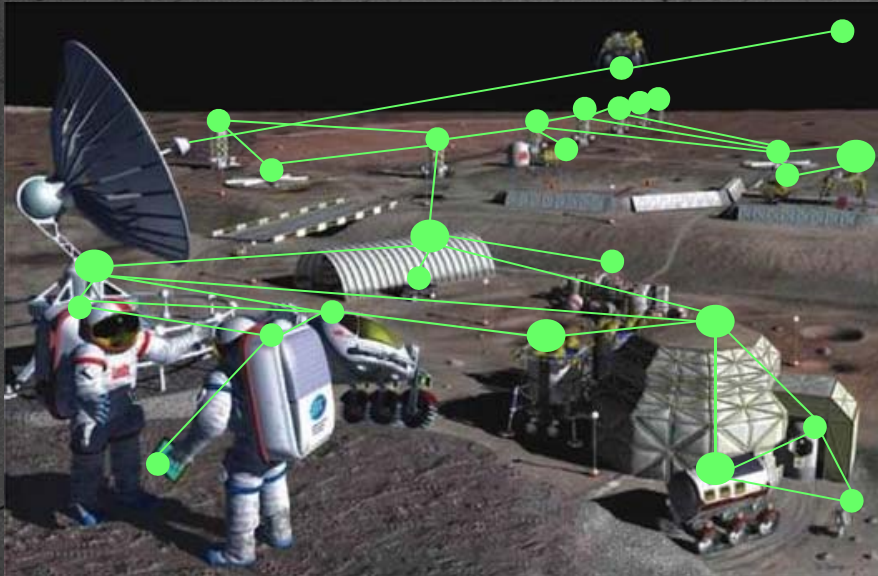
Internal Payloads ,
Free-flyers



Mesh vs. Separate Infrastructure



- Price, and Agility





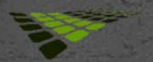
Mesh Protocols

- Many choices:
 - OLSR, AODV, BATMAN, BABEL, HWMP, CJDNS
- Reactive, Proactive, Hybrid
- Protocol Overhead, Radio Awareness, Layer 2 vs. 3
- Two, already in the Linux kernel:
 - B.A.T.M.A.N-Advanced, because a strong developer community has ported it to many platforms
 - 802.11-2012 (absorbed 802.11s, uses Hybrid Wireless Mesh Protocol), because it is the IEEE WiFi standard

Simplified Consumer Packaging



- Software



airmesh
instant mesh networking made easy



Byzantium



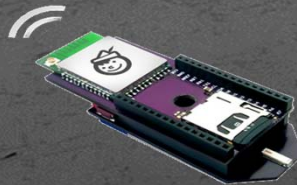
COMMOTION



Project Meshnet

- Hardware

pinoccio



Platforms



Netbook

- ASUS brand laptop
- Ubuntu 12.10 Linux
- Compat Wireless package
- “iw” wireless extensions package
- “batctl” package
- batman-adv.ko kernel module

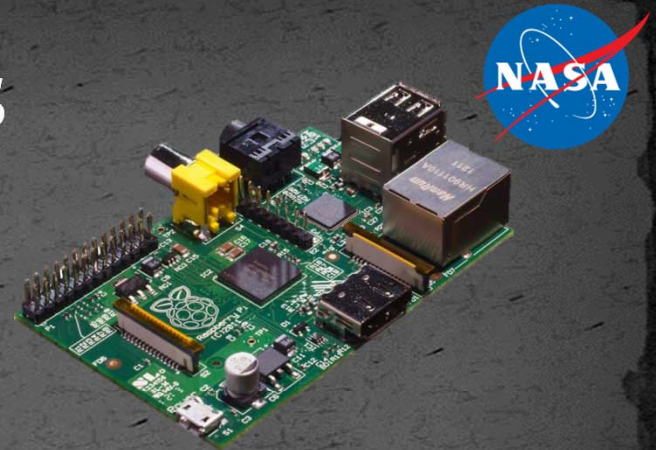
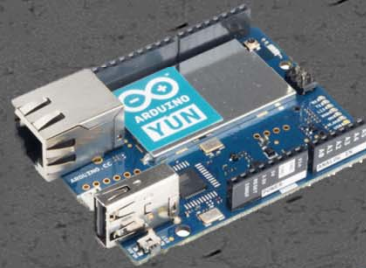


Wireless Access Point

- TP Link 1043ND
- OpenWRT Linux firmware
- “kmod-batman-adv” package

Other Popular Platforms

- **Raspberry Pi** – been done
- **Arduino** – been done
- **Android** – unsanctioned
- **Windows** – abandoned

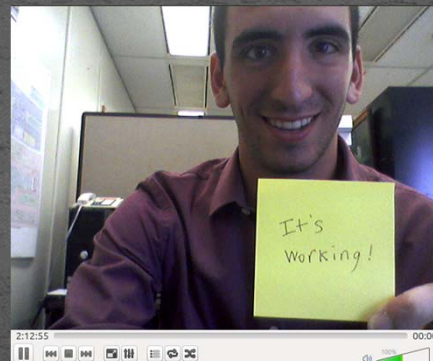
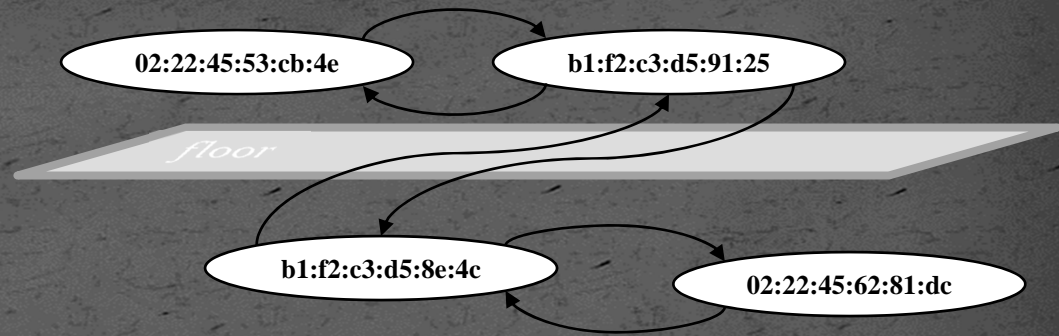


Windows 7
Microsoft Research



Video/Audio Streaming Test

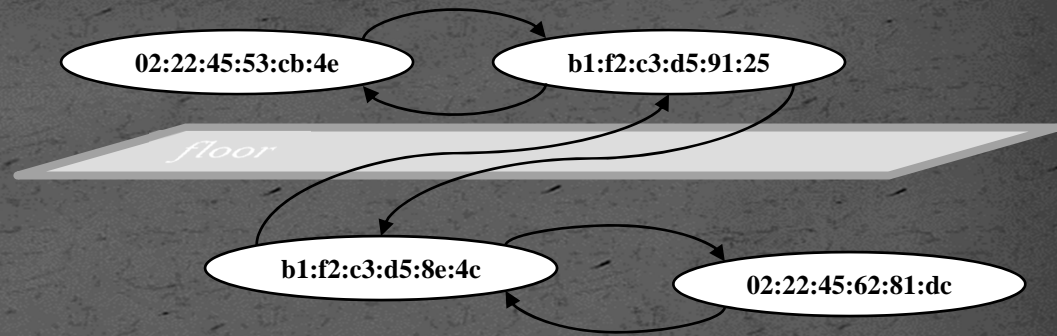
- Ekiga VoIP: dropped calls when routes switched
- VLC offered good audio and usable low-res video. High-resolution video streams did not tolerate dropped packets
- BATMAN or 802.11s





Throughput with Bottleneck

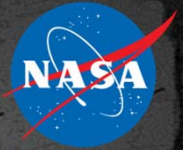
- Co-located node could transfer more than 50Mbps
- Penetrating the floor created a bottleneck of 10Mbps
- Additional hops had no cost



- BATMAN or 802.11s

Antenna Range Config

- Nodes equally spaced on 750m antenna range

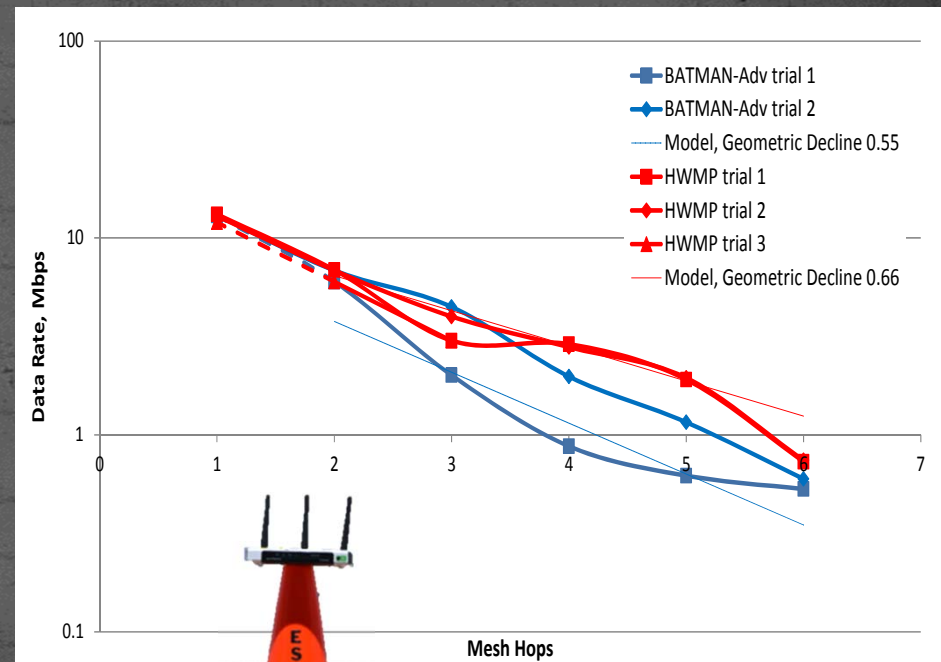


- BATMAN vs. 802.11S
- Throughput, and Mobility



Throughput with Equal Hops

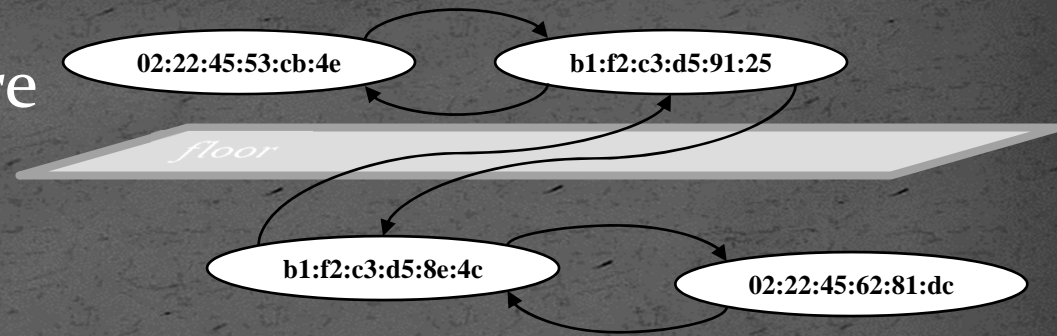
- Capacity dropped 34% - 45% per hop
- Independent of antenna geometry
- BATMAN or 802.11s





Route Switch with Node Failure

- Instantaneous failure by removing power from intermediate node
- Typical recovery in under a second
- Convergence could take 2-3 seconds

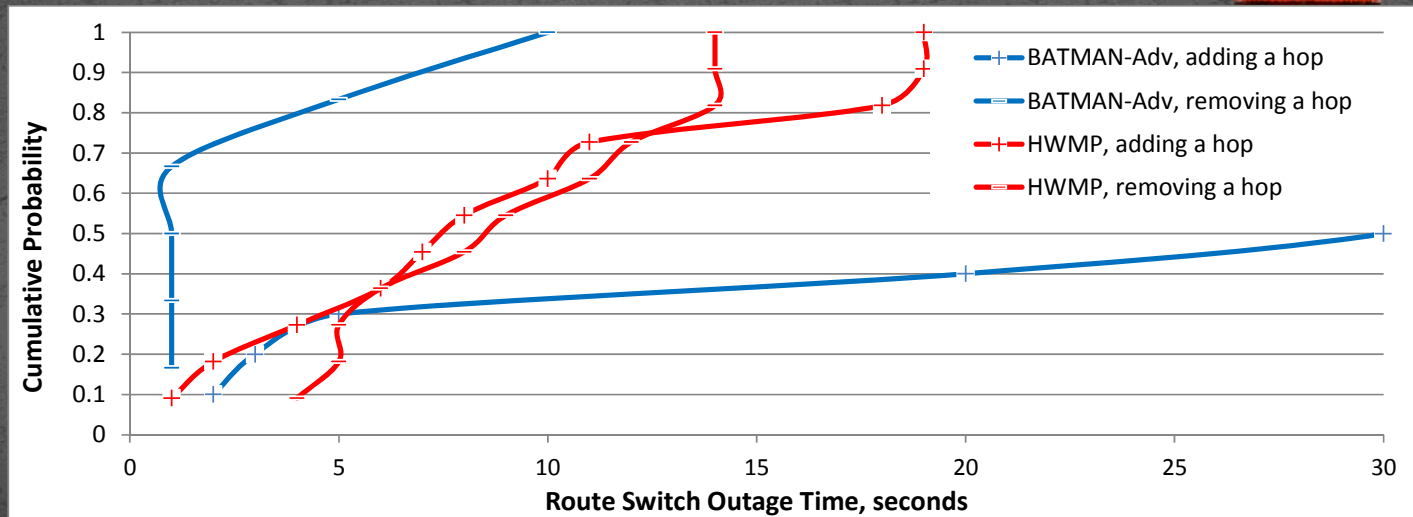


- BATMAN or 802.11s

Route Switch with Fading



- 802.11s had 7 to 20 seconds outage
- BATMAN-Adv can take minutes to add hops





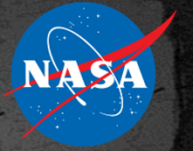
Mesh Security

- Wpa_supplicant (AES, TKIP) does not support Ad-hoc networking
- Only available option is using WEP encryption.
WEP has been broken since 2001, even using a 256 bit key it can still be cracked within minutes.
- Alternative Options:
SSH, VPN, DTN BSP, HTTPS



Other future work

- Visualization: Now built in to BATMAN but not 802.11s implementations.
- QoS optimization and management
- Relieving gateway congestion: geometry and protocol
- Multi-channel network
- Dynamic IPv6 addressing
- Wifi does not adapt transmit power
- Future technologies: 802.11ad, 4G LTE Advanced, CCSDS Proximity





Types of Nodes

- **Mesh Point:**
 - Every mesh node is a Mesh Point.
 - Can dynamically find neighbors and can route data to other Mesh Points.
- **Mesh Portal:**
 - Gateway to an external network.
- **Mesh Access Point:**
 - Access Points for clients to connect to the mesh.
- **Mesh Client:**
 - Attached to a Mesh Access Point as an access point
 - Cannot perform the mesh protocol



Routing Protocols

- There are three type of routing protocols:
 - **Reactive:** Search for a path between nodes when there is data to send
 - **Proactive:** Actively establish and maintain data paths no matter if data is being sent
 - **Hybrid:** Mix of Reactive and Proactive

Common Mesh Routing Protocols

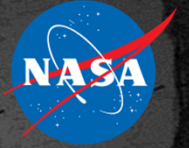


- IEEE Standard
 - 802.11s Working Group
 - HWMP (Hybrid Wireless Mesh Protocol)
- Other Development
 - OLSR (Optimized Link State Routing Protocol)
 - B.A.T.M.A.N (Better Approach To Mobile Adhoc Networking)
 - Babel (A loop-avoiding distance-vector routing protocol)

Common Mesh Routing Protocols



| Routing Protocol | HWMP | OLSR | Batman-Adv | Babel |
|-------------------|---------------------------|------------|------------|-----------------------|
| Reactive Routing | Between mesh nodes | None | None | Senses Route Failures |
| Proactive Routing | To mesh gateways | Everything | Everything | Almost Everything |
| Network Overhead | Low with minimal gateways | High | Medium | Medium |



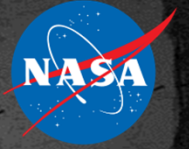
Routing Protocol Decision



Batman-adv and HWMP Chosen

- OLSR had too much network overhead
- Babel is the newest, didn't have support for a variety of devices, and had the smallest community base
- HWMP is the IEEE 802.11 standard
- Batman-adv is also in the Linux kernel and had support for a wide variety of devices plus a large community





Evaluation



Test Equipment

- Asus Netbooks (Ubuntu)



- TP-Link WR1043ND Routers (OpenWRT)



OpenWrt | OpenWrt Backfire 10.03.1-rc5 | Load: 0.00/0.00/0.00 | Changes: 0 Administration

System Services Network Logout
Interfaces DHCP and DNS Services Static Routes Firewall Diagnostic

Firewall

The firewall creates zones over your network interfaces to control network traffic flow.

General Settings | Custom Rules

Enable SYN-flood protection

Drop invalid packets

Input: accept

Output: accept

Forward: reject

Zones

| Zone | Forwardings | Input | Output | Forward | Masquerading | NAT clamping |
|------|-------------|--------|--------|---------|-------------------------------------|-------------------------------------|
| lan | lan | accept | accept | reject | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| wan | (empty) | reject | accept | reject | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |

Redirections

| Name | Protocol | Source | Via | Destination | Action | Sort |
|--------------------------------------|----------|--------|-----|-------------|--------|------|
| This section contains no values yet. | | | | | | |

Rules

| Name | Family | Protocol | Source | Destination | Action | Sort |
|------|---------------|---------------------|---------------|-------------------|--------|------|
| - | IPv4 only | UDP | wan:0.0.0.0/* | Device:0.0.0.0/68 | ACCEPT | + |
| - | IPv4 and IPv6 | ICMP (echo-request) | wan:0.0.0.0/* | Device:0.0.0.0/* | ACCEPT | + |

Reset Save Save & Apply

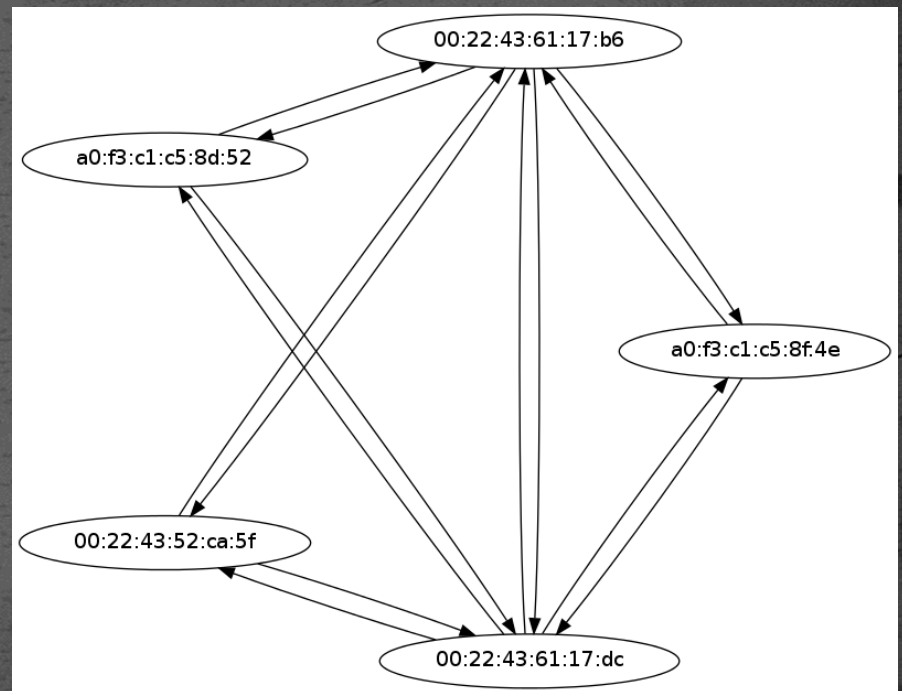
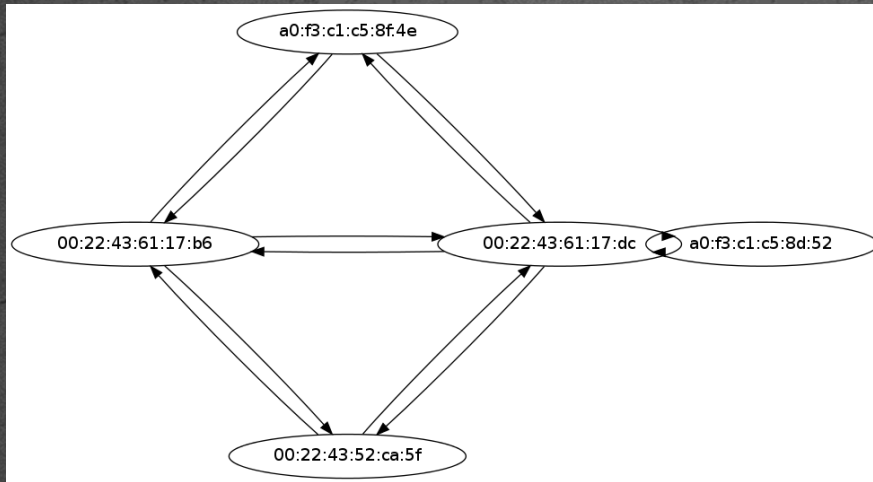
Powered by LuCI 0.10 Branch, Ev0.10+svn7292



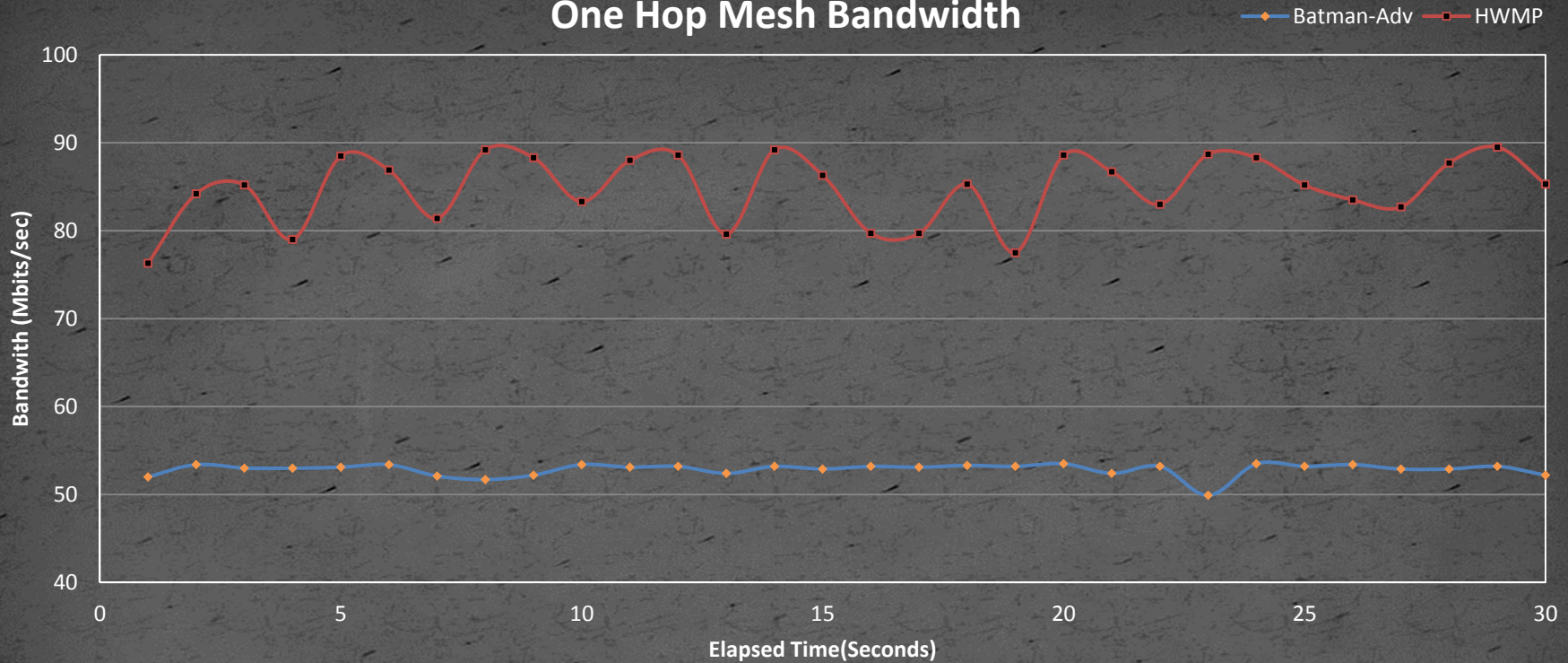
Preliminary Work

- HWMP already in the Linux Kernel
- HWMP doesn't have built in visualization, wrote scripts to do visualization.
- Batman-adv has an OpenWRT repository download.
- Batman-adv only has an Ubuntu repository download for *batctl* the control interface. Needed to compile the kernel modules myself.
- Wrote out installation and set-up instructions for both.
- Wrote start-up scripts for both the laptops and the routers.

General Set-up and Testing



One Hop Mesh Bandwidth



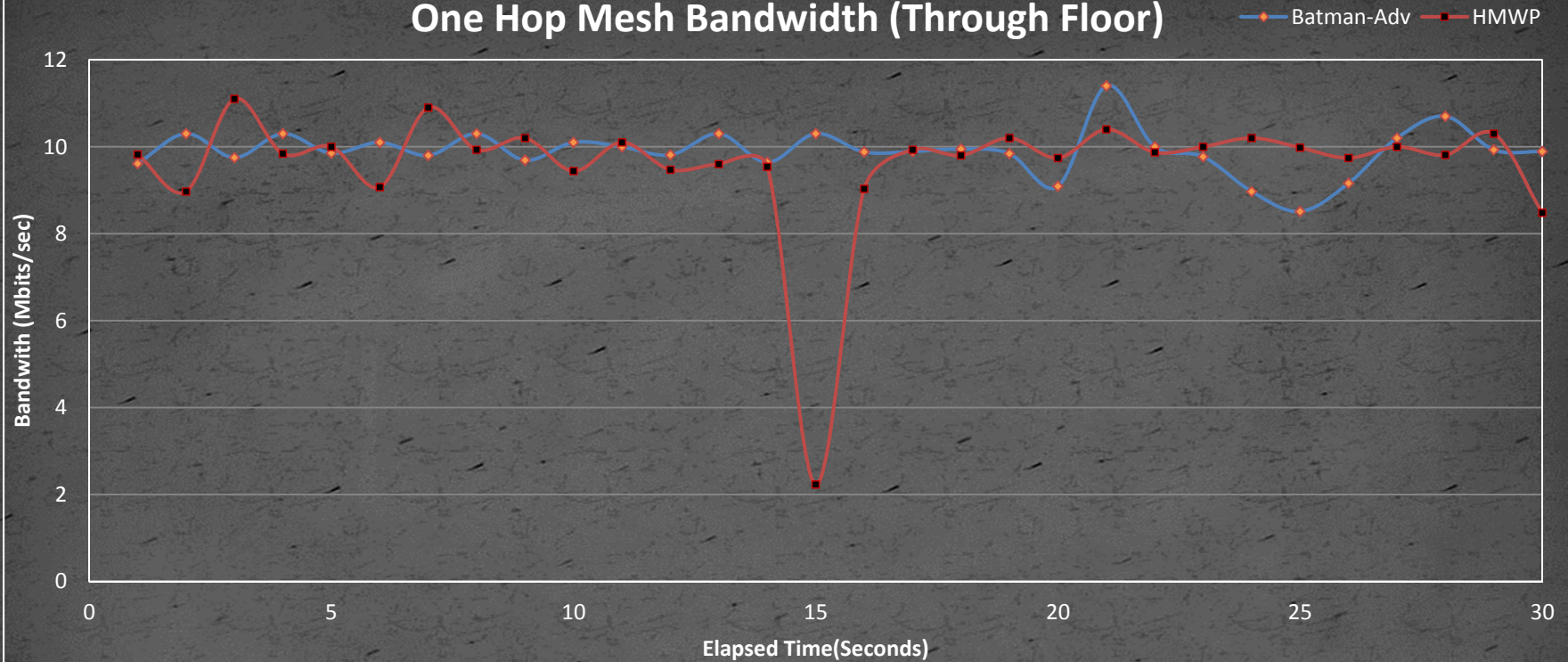
Batman-Adv

Average Bandwidth: 52.8 Mbits/sec
Average Jitter: 0.306 ms
Total Lost Packets: 1211/136047 (0.89%)

HWMP

Average Bandwidth: 84.2 Mbits/sec
Average Jitter: 15.457 ms
Total Lost Packets: 2189/218689 (1%)

One Hop Mesh Bandwidth (Through Floor)



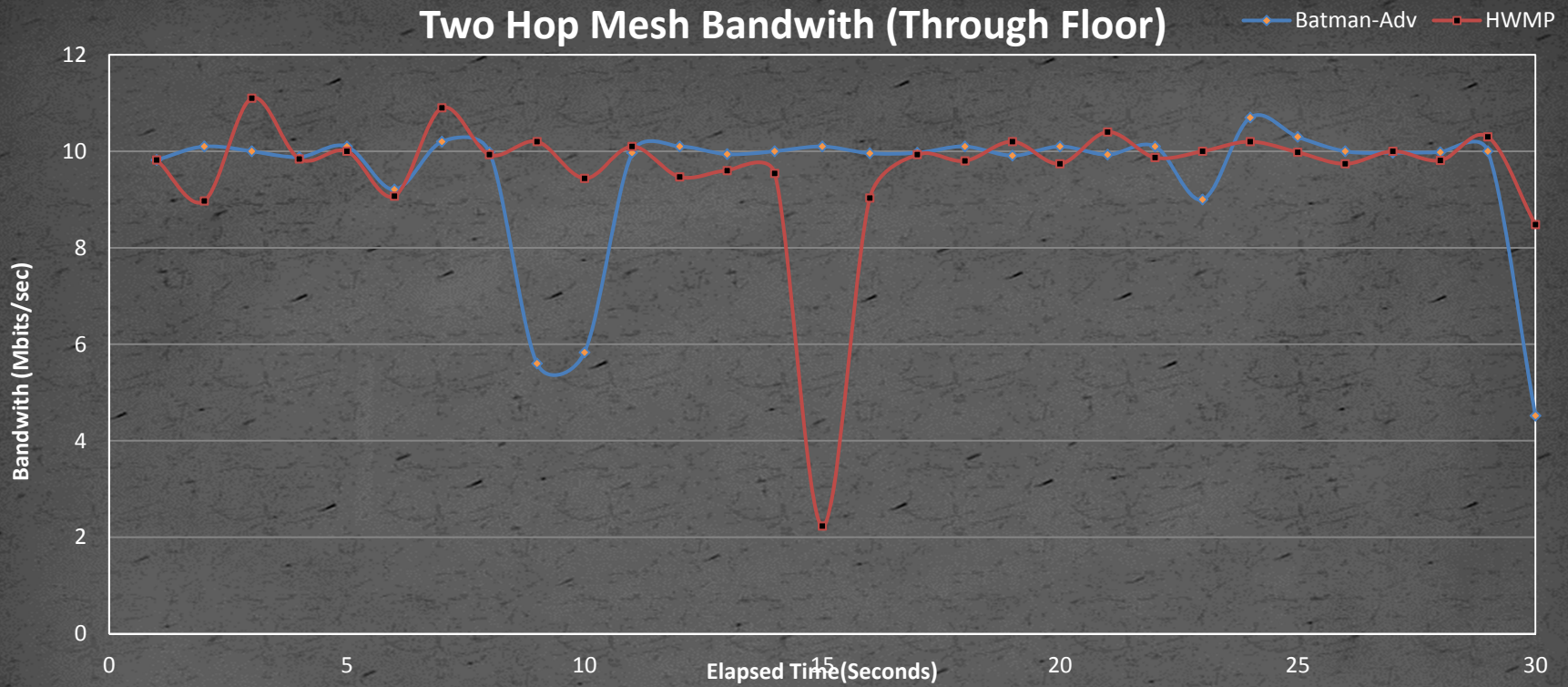
Batman-Adv

Average Bandwidth: 9.91 Mbits/sec
Average Jitter: 1.444 ms
Total Lost Packets: 225/25512 (0.88%)

HMWP

Average Bandwidth: 9.34 Mbits/sec
Average Jitter: 13.806 ms
Total Lost Packets: 227/24254 (0.94%)

Two Hop Mesh Bandwith (Through Floor)



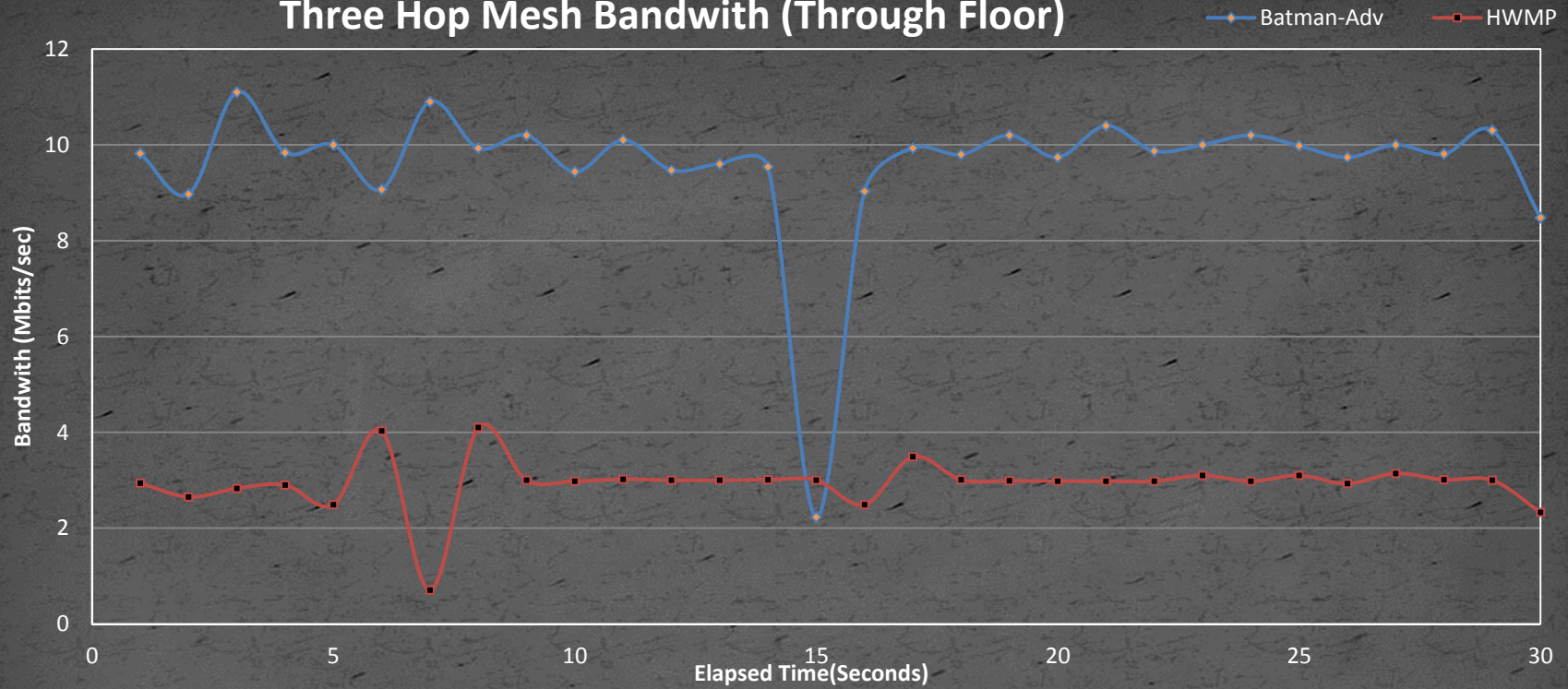
Batman-Adv

Average Bandwidth: 9.35 Mbits/sec
Average Jitter: 25.605 ms
Total Lost Packets: 1134/25512 (4.4%)

HWMP

Average Bandwidth: 8.55 Mbits/sec
Average Jitter: 1.443 ms
Total Lost Packets: 1146/22971 (5%)

Three Hop Mesh Bandwith (Through Floor)



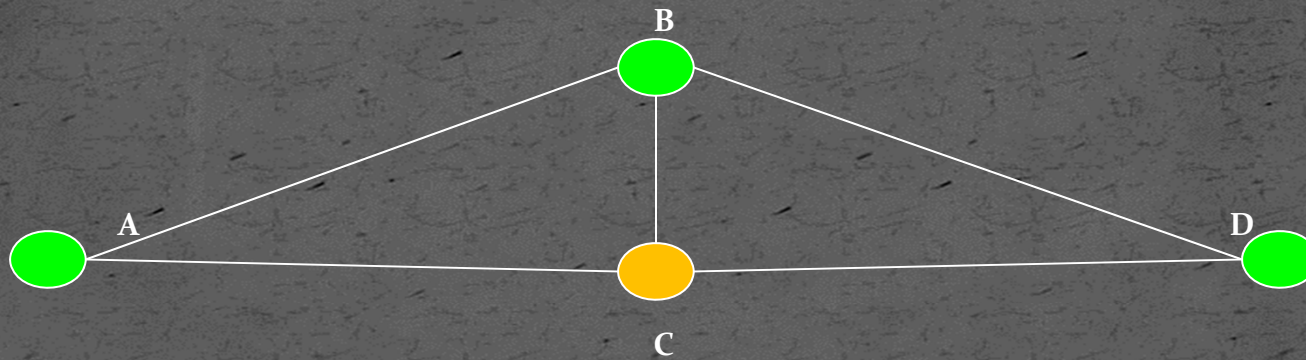
Batman-Adv

Average Bandwith: 9.61 Mbits/sec
Average Jitter: 1.328 ms
Total Lost Packets: 909/25510 (3.6%)

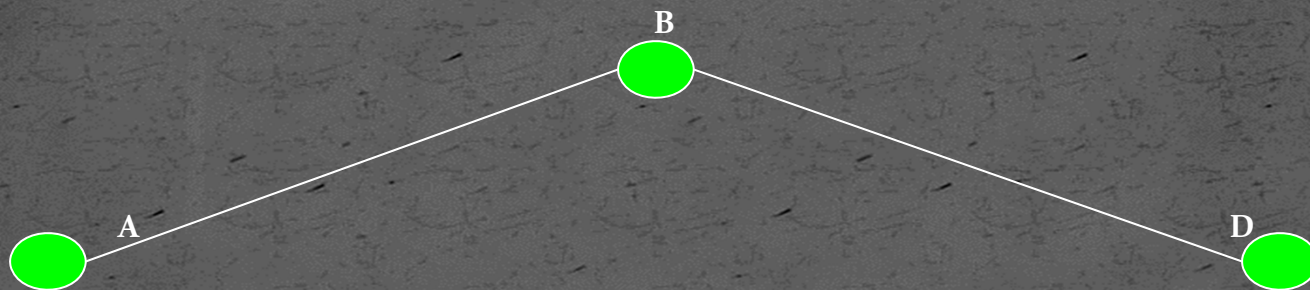
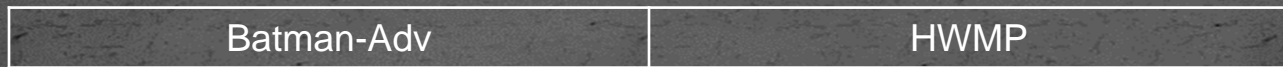
HWMP

Average Bandwith: 2.95 Mbits/sec
Average Jitter: 2.842 ms
Total Lost Packets: 99/ 7652 (1.3%)

Convergence Time



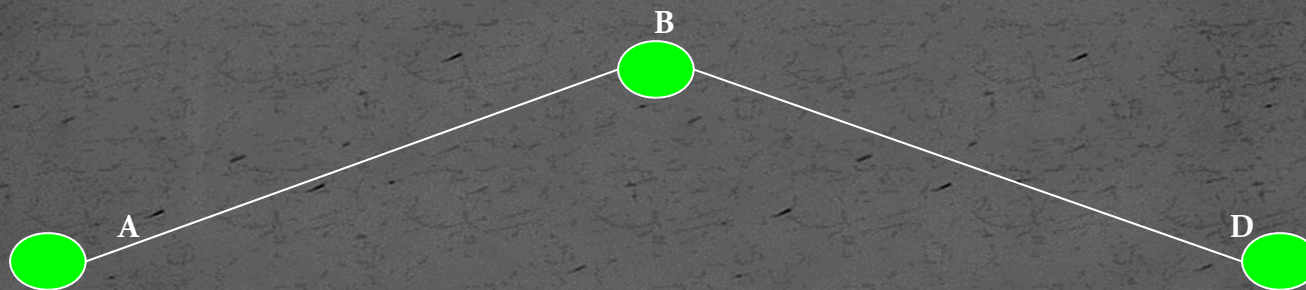
Convergence Time



Convergence Time



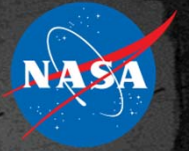
| Batman-Adv | HWMP |
|----------------|----------------|
| Low: 120ms | Low: 140ms |
| Average: 743ms | Average: 443ms |
| High: 2120ms | High: 2660ms |



Antenna Range Testing



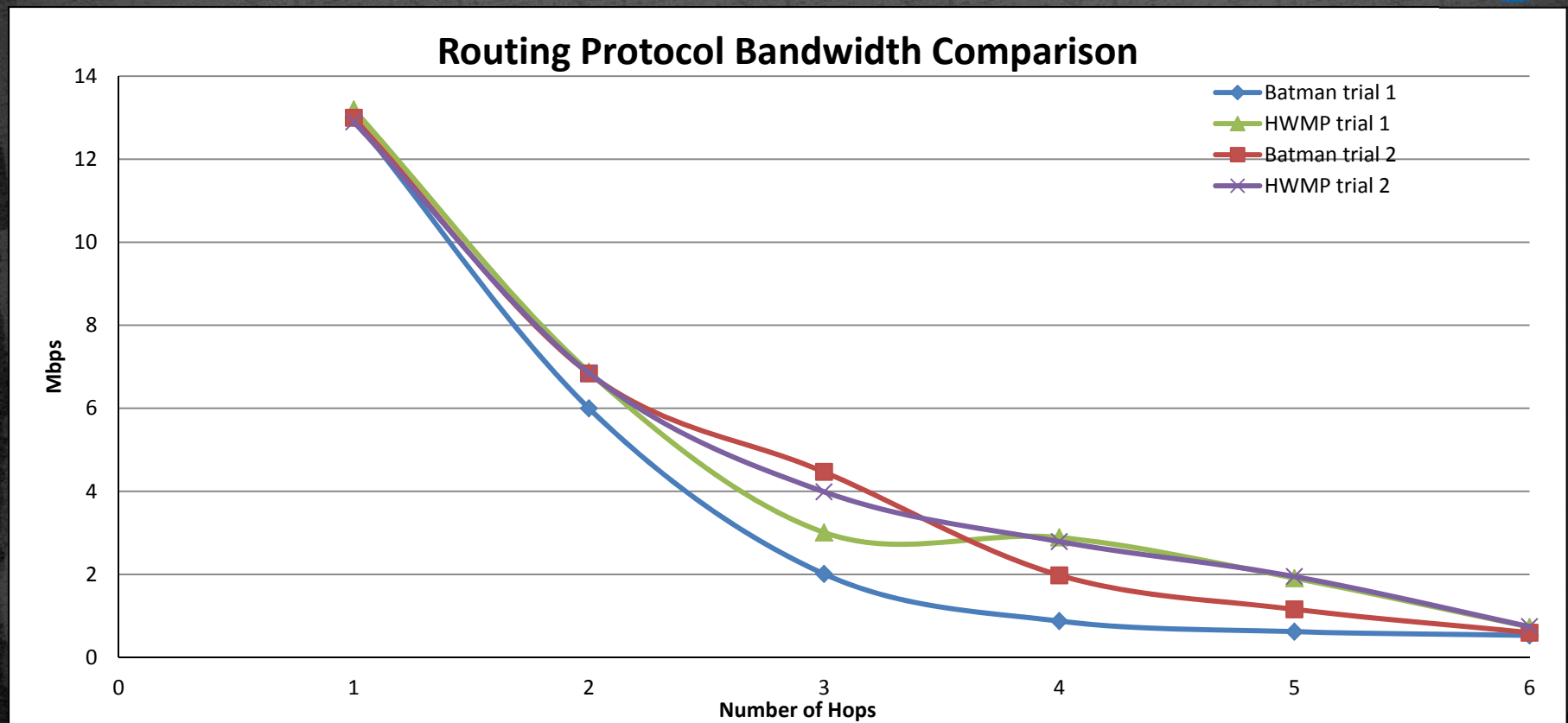
Antenna Range Testing



Paces Between Nodes



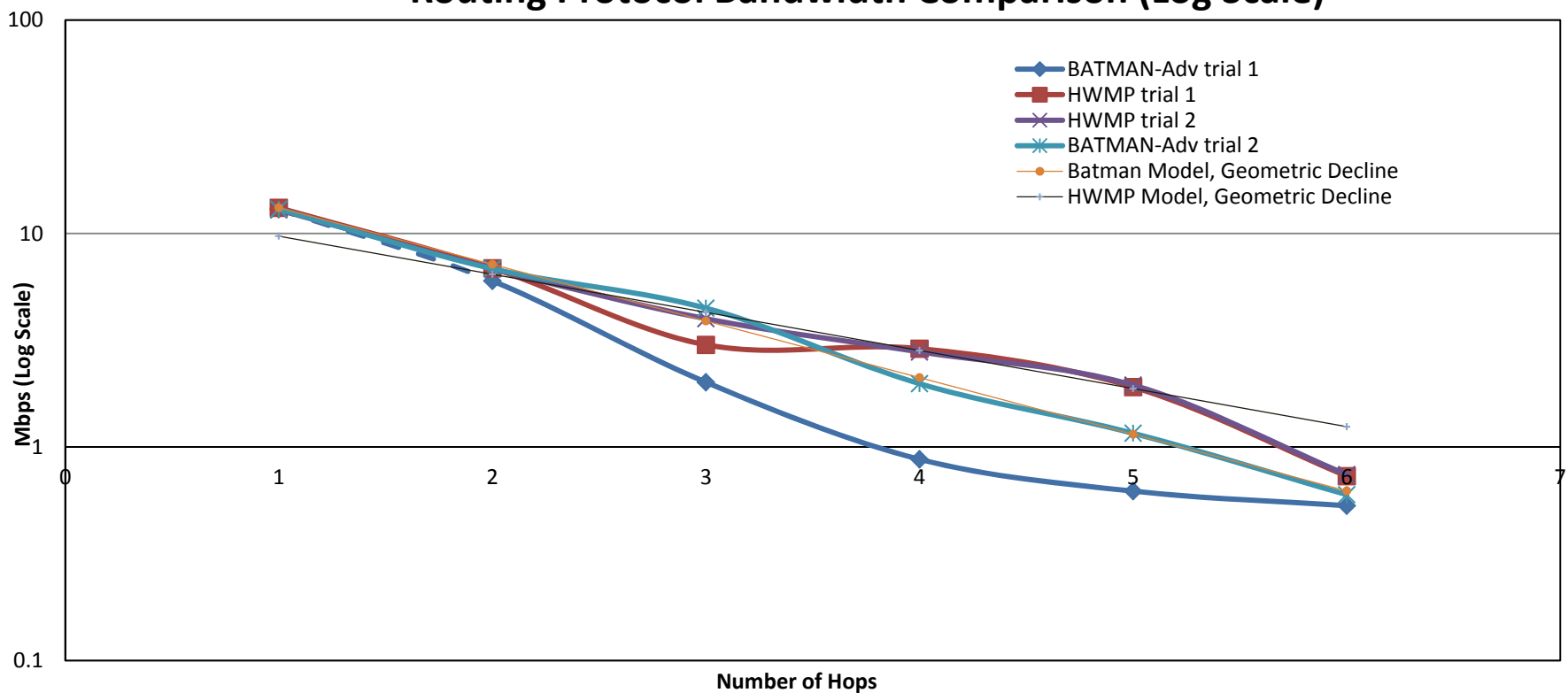
Antenna Range Results



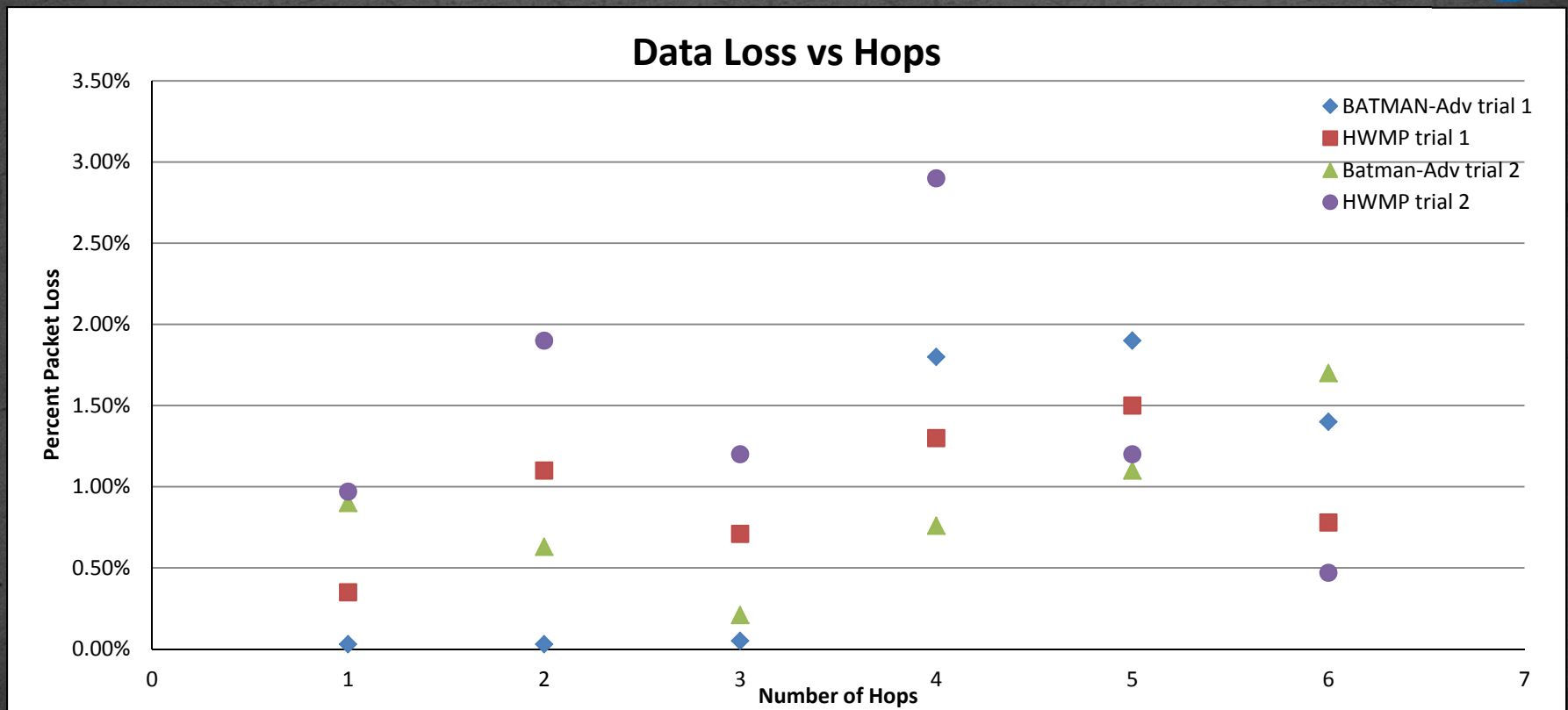
Antenna Range Results



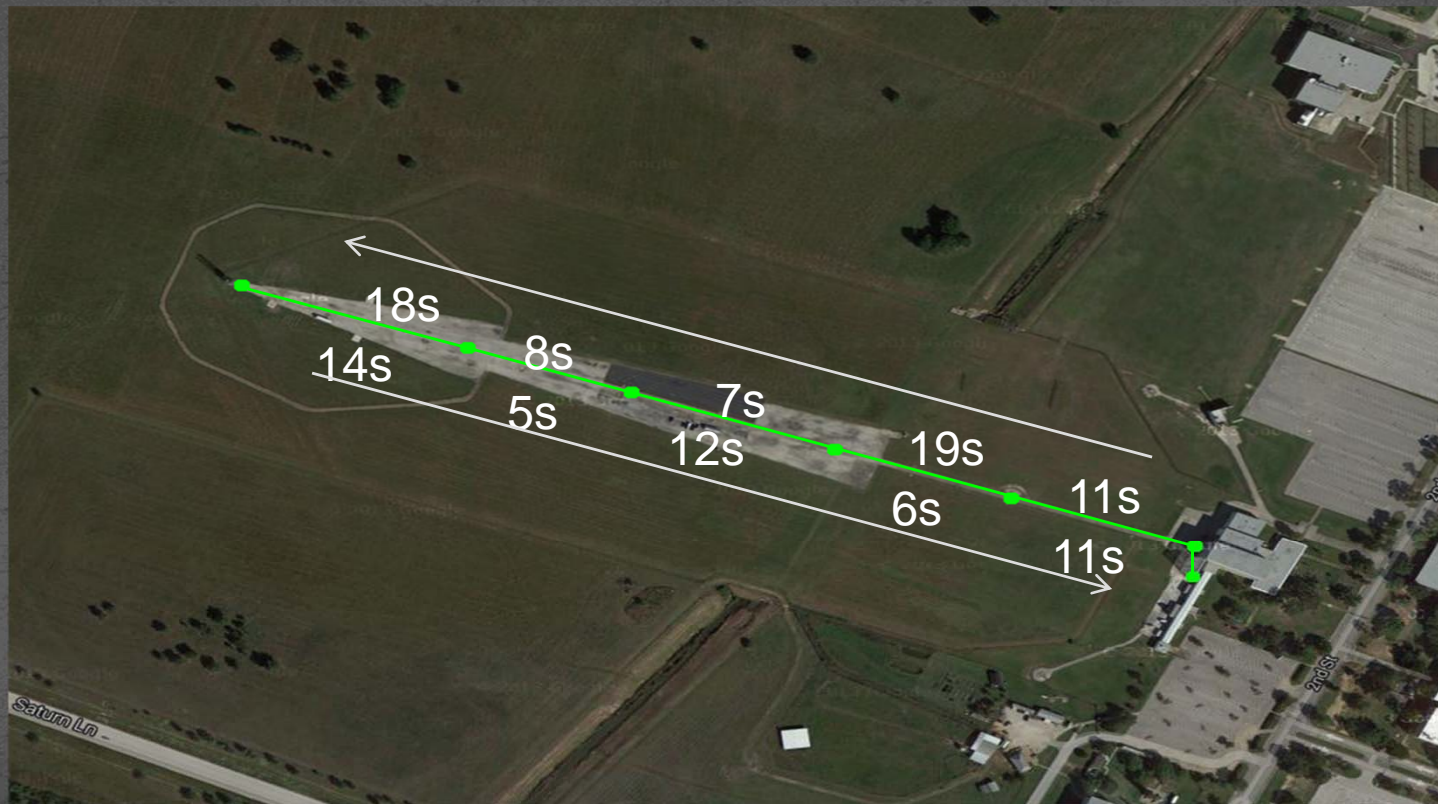
Routing Protocol Bandwidth Comparison (Log Scale)



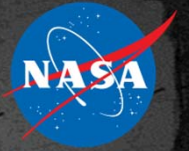
Antenna Range Results



HWMP Handoff Times



Batman-Adv Handoff Times



Test Statistics



Batman-adv

45% Bandwidth decrease per hop

.87% Average data loss

5.25s Average handoff moving **towards** node

3m58.5s Average handoff moving **away** from node

HWMP

34% Bandwidth decrease per hop

1.29% Average data loss

12.6s Average handoff moving **towards** node

9.6s Average handoff moving **away** from node

Test Statistics



Batman-adv

45% Bandwidth decrease per hop

.87% Average data loss

5.25s Average handoff moving towards node

3m58.5s Average handoff moving away from node

HWMP

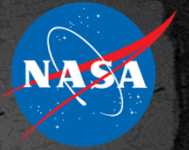
34% Bandwidth decrease per hop

1.29% Average data loss

12.6s Average handoff moving towards node

9.5s Average handoff moving away from node

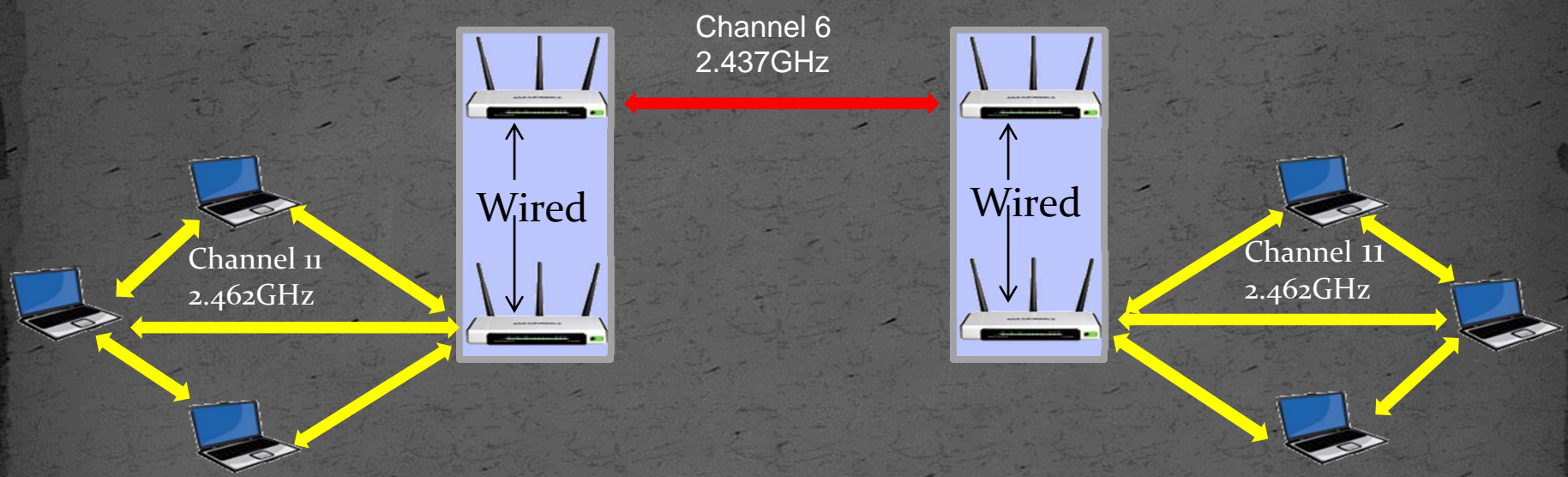
Favored



Other Areas of Interest



Multi Frequency Mesh



Multi Frequency Mesh



Channel 1
2.412GHz



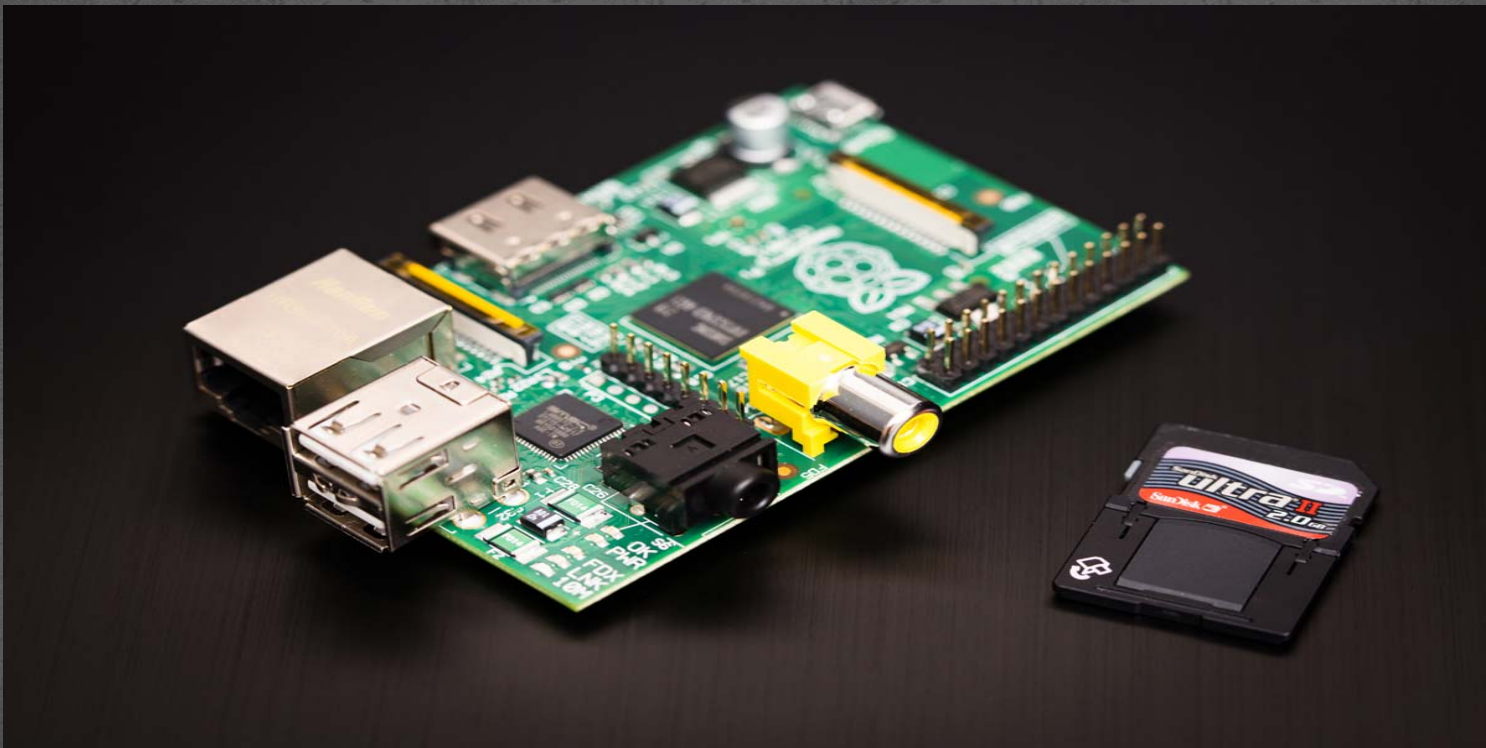
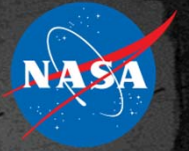
Channel 6
2.437GHz

Channel 11
2.462GHz

Channel 11
2.462GHz



Raspberry Pi's



Quality of Service and DTN



OpenWrt | OpenWrt Attitude Adjustment 12.09-rc1 | Load: 0.14 0.05 0.05

Changes:

Status System Services **Network** Logout

Interfaces Wifi Switch DHCP and DNS Hostnames Static Routes Firewall Diagnostics **QoS**

Quality of Service

With QoS you can prioritize network traffic selected by addresses, ports or services.

Interfaces

WAN [X] Delete

| | |
|-------------------------|---|
| Enable | <input checked="" type="checkbox"/> |
| Classification group | default |
| Calculate overhead | <input checked="" type="checkbox"/> |
| Half-duplex | <input type="checkbox"/> |
| Download speed (kbit/s) | 25600 |
| Upload speed (kbit/s) | 2048 |
| | <input type="text"/> <input type="button" value="Add"/> |

Classification Rules

| Target | Source host | Destination host | Service | Protocol | Ports | Number of bytes | Sort | |
|----------|-------------|------------------|---------|----------|-----------------------------|-----------------|------|------------------------------------|
| priority | all | all | all | all | 22,53 | | ↑ ↓ | [X] Delete |
| normal | all | all | all | TCP | 20,21,25,80,110,443,993,995 | | ↑ ↓ | [X] Delete |
| express | all | all | all | all | 5190 | | ↑ ↓ | [X] Delete |
| low | all | all | all | all | 9090 | | ↑ ↓ | [X] Delete |
| | | | | | | | | <input type="button" value="Add"/> |

What's in the Box?

- Configurable virtual components of a router

