ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2017

Multi-hop Mobile Wireless Mesh Network Testbed Development and Measurements

Paulo Alexandre Regis, Cayler Miley, Shamik Sengupta

Department of Computer Science and Engineering University of Nevada, Reno, USA, 89557

E-mail: ssengupta@unr.edu

Abstract: Mesh networks are envisioned to play a major role in the future of Smart-Cities and Internet of Things technologies, and can be a great ally in executing mission-centric tasks like search and rescue, and assist first responders. In this paper, we demonstrate the feasibility of development of a mobile mesh network testbed with the help of commercial-off-the-shelf (COTS) components and perform measurement tests in an indoor environment. We used open-source software to enable the testbed, such as Linux as the operating system, and Optimized Link State Routing (OLSR) protocol to enable the wireless mesh network. The physical components include open-source hardware like Raspberry Pi computers, and mobility is enabled through the use of a mobile robot Pioneer 3. We evaluate the network in various scenarios, first establishing benchmark measurement in a stationary setup, and then comparing with a mobile variation.

Keywords: Mobile mesh; Ad hoc network, Testbed, OLSR, Measurement, Multi-flow

I. INTRODUCTION

Multi-hop mobile mesh networking is anticipated to be the solution to the ever increasing demands placed on cognitive wireless networks to accommodate growing First responders' networks, disaster area networks, Military Tactical Networks and Commercial applications. Unlike infrastructure networking, multi-hop point-to-point/mesh architecture can create wide-area cognitive backhaul networks where traffic can flow among the peers directly using relay/forwarding via multi-hops creating the possibility of higher capacity, ubiquitous connectivity and increased coverage.

Recently, cognitive mobile mesh networks have drawn significant attention from researchers in academia and industry due to the advancement in technologies and applications such as Internet-of-Things (IoT), smart-cities, traffic control, wearable devices, health monitoring, etc. Mobility is one of the most important inherent features of such multi-hop networks, be it first responders' networking, disaster area networking or commercial human mobile networking. Despite its importance, however, mobility is still hardly explored in the context of cognitive radio and dynamic mesh networks. Allowing cognitive radio node mobility in multi-hop mobile mesh networks will introduce numerous challenges, making it necessary to revisit current system design and protocols, such as mechanisms for neighbour discovery, dynamic routing, spectrum sensing, interference management, dynamic spectrum access, and many more. However, currently, there is little understanding of how such a cognitive architecture will operate so as to make the system feasible under mobility and multi-hop paradigms. An example network is depicted in Figure 1.

Although cognitive mobile mesh networks have the potential to achieve efficient spectrum usage, interference mitigation, higher throughput and extended coverage, there are several open research issues in the design of multi-hop mobile mesh networks that need to be addressed first before they can be successfully deployed. Firstly, mobile multi-hop mesh networks still face all the challenges that are inherent in legacy wireless ad hoc networks. For example, devices are inherently heterogeneous in nature due to diversity in wireless access technologies, user terminals, types of services and applications as well as the spectrum bands. Apart from these, multi-hop mobile mesh design must deal with additional challenges that arise from the mobile nature of the communications. For example, it is not just sufficient to identify communication channels but to also regulate resource sharing among all flows in the network and more importantly accomplish efficient end-to-end communication by establishing routing paths amidst the dynamically changing sets of mobile cognitive nodes.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2017

For legacy infrastructure wireless networks, the mobility management challenges basically focus on address management and configuration, paging, network architecture maintenance, registration with new gateways/homes, tunnelling to selected gateway(s), and handover decision [1]. Most of the existing research on such mobility management divides the problems into several micro-mobility management problems by splitting (logically or physically) the entire coverage area into multiple smaller networks to generate intra-domain and/or inter-domain mobility management solutions.

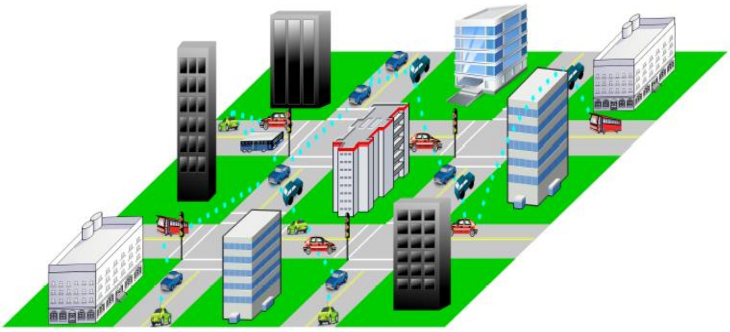

Figure: 1 Mobile mesh network communication example.

However, in cognitive multi-hop mobile mesh networks, as a user moves geographically, the mobility management challenges include not only frequent location changes but also immediate neighbour discovery, dynamic routing, spectrum sensing, interference management, and rendezvous with already communicating nodes. This variation will also depend on the mobility models of the individual nodes. Hence, the space-time variation in the availability of opportunities as well as other conditions will affect network design and management in multi-hop mobile mesh networks. Accordingly, the challenges in the design of cognitive multi-hop mobile mesh networks must be better understood for the concept of such networks to reach its full potential [2].
Research efforts in this area can be seen in [3-7] and the references therein. However, they rely mainly on simulation to validate new models. Although an important step towards enabling its application, simulations cannot account for every variable the real world may have. To provide insights and grasp of how a new approach or mechanism may perform, the deployment of such ideas in a prototype manner becomes necessary. Thus, the development of testbed systems to allow performance evaluation and feasibility testing is equally important as simulations.
Building specialized hardware to test novel approaches can be very expensive. Thus generic hardware parts that provide flexibility and customization are good options for testbed architectures [8]. For example, Software Defined Radios (SDR) [9] can be useful when testing medium access techniques, as this allows transmission and reception in a wide range of the frequency spectrum. Open-source hardware initiatives, such as Arduino [10] and Raspberry Pi [11] can be viable options to develop cost effective robots and controllers. Raspberry Pi is a simple credit-card sized computer running a Linux operating system. It provides flexible customization even at the kernel level, allowing testing of new network protocols and hardware drivers in a widely used environment.
In this paper, we investigate and demonstrate how to build a simple mobile mesh network testbed with commercial off-the-shelf (COTS) components, such as Raspberry Pi computers, Pioneer 3-DX ground robots, and Logitech gamepad controller. We also make use of open-source software available to the community. We automate the setup process and enable fast deployment of the software. The source code and instructions can be accessed at [12]. The open-source initiative allows fast prototyping of new features and fine tuning of parameters in routing protocols. Our system is based on Linux operating system and uses the optimized link-state routing (OLSR) protocol to enable the mesh network.
The network constitutes of a combination of fixed and mobile nodes, representing the mobile diversity possible in such a network. Using this testbed, we can perform a set of experiments and measure the performance of the network in various cases. We can measure any network parameters, such as throughput, delay, and jitter, in the mobile setup. The building process and experimentation provide a first-hand experience of how such network can perform under different conditions, and how to physically build a testbed to test future novel approaches.
The remainder of the paper is organized as follows. In Section II we explore the literature and other works related to the system development. Section III gives a description of our system. In Section IV we explain the set of experiments

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2017

devised to evaluate the performance of the system and discuss their results. Finally, Section V concludes the paper and suggests future research directions.

II. RELATED WORK

In recent years, researchers have focused on applications that would make use of the future mobile mesh concept. For example, WW Jan, et al. [13] proposes a traffic congestion recognition and avoidance algorithm where vehicles transmit their velocity to the neighboring cars. Based on the received data, the navigation system can reroute the path to avoid slow traffic areas. E Nnanna, et al. [14] propose an incident warning application (IWA), in which the vehicles adapt their routes based on reports from other vehicles. T Gregor, et al. [15] proposes a mission aware topology management algorithm that considers various mission parameters from the entire network to choose which node(s) should relocate. K Murat, et al. [16] studies the application of mobile mesh network in smart grid monitoring in various different scopes, such as home network, and wide area network.

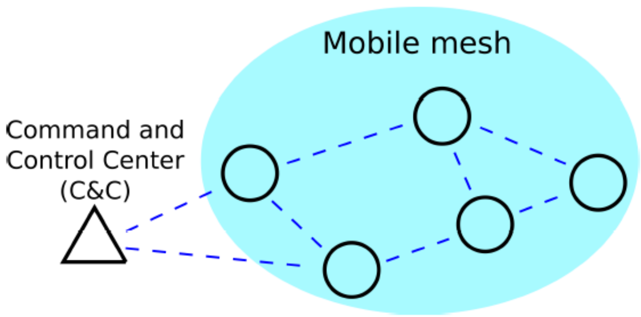

Figure 2: Network model.

The different characteristics of applications have a different requirement regarding communications. G Florian, et al. [17] compares inter-domain (i.e. communication between groups of mobile networks) routing approaches in battlefield scenarios. In military applications, mesh networks can play a major role. Unmanned Autonomous Vehicles (drones) are used in dangerous situations when sending manned aircraft is a high risk. Such cases may involve intelligence, reconnaissance and surveillance applications, providing communications to temporarily deployed troops, and remote monitoring of environmental conditions [3].

Comparatively, there are much fewer works available where researchers are exploring the physical viability of developing inexpensive mesh network prototype for testing novel research issues in different conditions. H Songfan, et al. [18] build a testbed using Personal Digital Assistants to create a mesh network and compare two distinct routing protocols, namely Ad hoc On demand Distance Vector (AODV), and Dynamic Source Routing (DSR), in both indoor and outdoor scenarios. Work by D Johnson, et al. [19] evaluates the performance of different routing metrics, hysteresis and expected transmission count (ETX), in a fixed position 7 by 7 grid testbed using OLSR protocol and conventional computers. Although ETX is a more sophisticated metric, authors identify cases in which the simpler hysteresis outperforms ETX. A Banerjee, et al. [20] Phantom Net is a testbed developed at the University of Utah. It allows prototyping and testing on real devices in a controlled environment. It uses COTS components, and enables 5G experiments using state-of-the-art technologies. It provides tools to emulate handovers between base stations. Orbit Lab [21] is a pioneering testbed with a vast number of nodes. The hardware has been updated throughout the years, and currently focuses on WiFi, WiMAX, and LTE coexistence. B Admir, et al. [22] combines open-source software and low-cost hardware and perform indoor experiments comparing different operating systems that enabled OLSR mesh networks. Their set up was made up of static nodes in line-of-sight positions, and a single traffic flow. These testbeds allow for a wide range of experimentation to be realized. However, physical movements of the network nodes is not a factor considered in the works above. This research paper is an attempt to bridge the gap between theory and practice.

III. SYSTEM DEVELOPMENT

To build our general purpose mobile mesh testbed, we borrow some concepts from real applications of such networks. The purpose of a mobile mesh network may vary, but there is always an objective to be achieved. For example, wireless sensor networks that measure some variables, such as humidity, smoke levels, always report to a control center, which will process the collected data and act accordingly. Mobile robots deployed in a disaster zone can look for signs of victims and report back to the command and control (C&C) center, when sending a human may be too dangerous.

In Figure 2 we see an example of such a network model. The C&C is a stationary node, and it serves as the information hub. The remaining nodes send all their traffic to the C&C. The traffic flow may or may not relay through multiple nodes to reach its destination.

Our system is built using commercial off-the-shelf (COTS) components. Using COTS proved to speed up the development of the testbed. Community support assisted in clearing the roadblocks faced, and reduced price of popular items enable larger scale systems. In addition to COTS components, we utilized open-sourced hardware and software when available, allowing in-depth customization if needed. Each part composing the testbed is an area of research by itself. Next, we describe each hardware and software component adopted in the building process of the testbed.

3.1 Optimized Link State Routing (OLSR):

The OLSR protocol [23], as its name suggests, maintains an updated status of the network link states. OLSR is flexible enough to work in both infrastructure and mesh modes; in this paper, we use the latter. OLSR distinguishes from other link state routing protocols because of its controlled method of broadcasting the network. Each node discovers its 2-hop neighbors through the dissemination of Hello messages. Then, individually, a node selects the Multipoint Relay nodes (MPR). The MPRs are selected in such a way there exists a path to the 2-hop neighbors through these MPR nodes. MPR nodes are responsible for forwarding Topology Control messages (TC). OLSR does not flood the entire network with all the link state information; rather it disseminates only the MPR selector information. Designing a flooding algorithm for ad hoc wireless networks can be tough, and by sending link state information only to the MPR nodes, it reduces the traffic load on the entire network. Doing so does not guarantee reliability in table synchronization. Thus, OLSR sends TC messages frequently enough to keep the databases somewhat synchronized.

3.2 Hardware:

Each node in the ad hoc network is composed of the hardware specified in Table 1. The use of Raspberry Pi computers allows future large-scale deployments due to its reduced cost. The open-source nature of these computers also enables greater support from the community. The wireless adapter is configured to work in 802.11g ad-hoc mode. All nodes are pre-configured to use the same channel of the IEEE 802.11 standard. To enable easy placement of the static nodes, and the mobility of the ground vehicle, 12 V batteries were used as the power supply.

Equipment	Description
Main processor	Raspberry Pi Model B
Storage	4 GB SD card
WiFi adapter	Realtek 8818eu
Power supply	12V 5Ah UB1250 battery
Power converter	12V to 5V USB step down converter
Mobile robot	Pioneer 3

Table 1: Hardware Parts List.

In Figure 3 we see the hardware that constitutes the mobile robot node assembled. The communication hardware (Raspberry Pi) and the mobile robot do not depend on each other to interact. In fact, the robot is controlled by a radio frequency controller. It is not a system limitation, the Raspberry Pi can directly interact with the robot, but in this work, it is not necessary as we need a more controlled setup.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2017

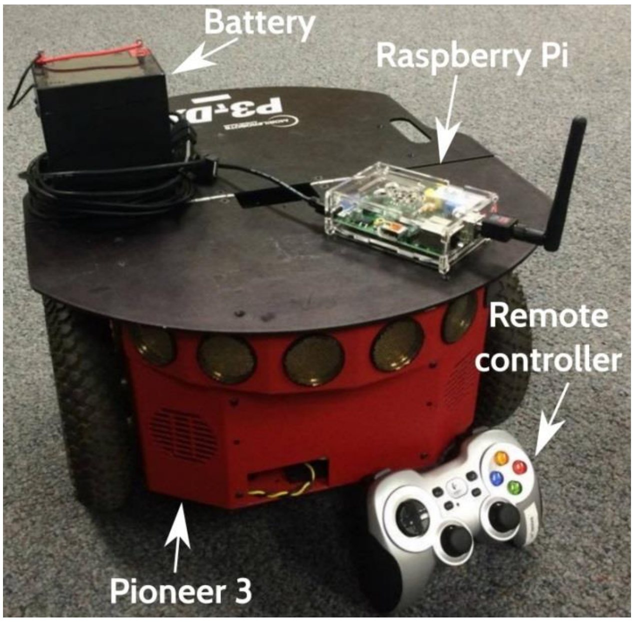

Figure 3: Mobile node hardware.

3.3 System software:

Table 2 shows a detailed description of the software utilized. We used the operating system provided by the Raspberry Pi foundation due to its plug and play capability. Since it is based on Linux kernels, it is suitable to research experiments and customization when necessary. We used the Optimized Link State Routing Protocol (OLSR) because of its open-source nature and extensive on-line documentation. We modified the source code to remove warnings at compilation time. We also developed an automated installation script to accelerate the deployment of new nodes. The source code and instructions can be accessed at [12]. In Table 3 the OLSR parameters values used are listed.

Software	Description
Operating system	Raspbian Jessie (Linux kernel 4.4.11 #888)
Link layer	IEEE 802.11g
Routing	OLSR
Measurement tools	Clink, Iperf, traceroute, ping

Table 2: Software List.

Parameter	Value
Daemon version	0.9.1
Mode	Mesh
Hello interval/validity	6 s / 600 s
TC interval/validity	0.5 s / 300 s
TC redundancy	2
MID interval/validity	10 s / 300 s
ETX metric	enabled
Poll rate	0.05 s
MPR coverage	5
Link quality level	2
Willingness	3

Table 3: OLSR Parameters.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2017

IV. EXPERIMENT AND RESULTS

We designed experiments to investigate different situations that can occur in a real world scenario. We start by measuring the performance in a static network, and progress to more complex setups with mobility and multiple flows.

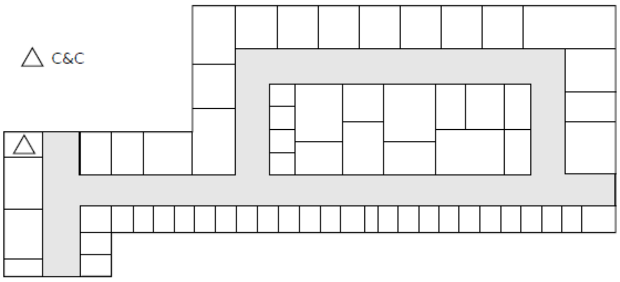

Figure 4: Floor plan diagram.

Figure 4 shows the floor plan where the experiments were performed. The white blocks are physical rooms. The content of the rooms are unknown, and may or may not interfere with the propagation of the signals, thus influencing the resulted measurements. The shaded area represents the hallways, where the nodes of the experiment are deployed. The triangle represents the C&C node, the starting point of all experiments.

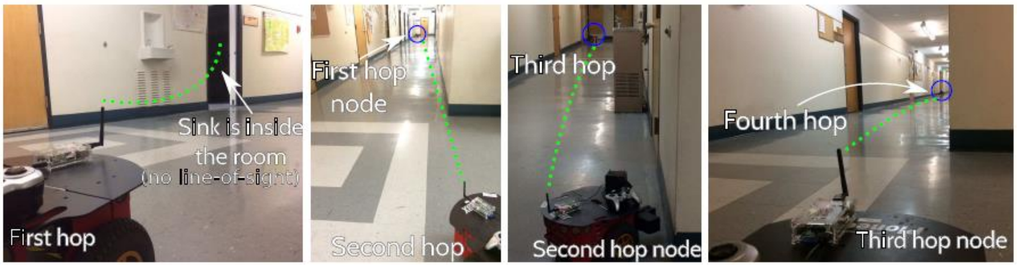

Figure 5: The testbed in action.

4.1 Static scenario:

The first set of experiments serves as a benchmark scenario. First, we define a feasible positioning for the static nodes. Then we measure the performance at each hop count (from 1 to 4). Lastly, we measure the performance when multiple flows are present.

1. Node deployment and path characterization:

In the first experiment, we estimate the characteristics of the network links. We start with an initial positioning guess and measure the characteristics of each link. Then we change the positioning and measure again. We iterate over the steps of placement and path estimation until a multi-hop setup with similar link characteristics is achieved. The path characterization is performed using the Clink tool [24] that implements the statistical method described by DB Allen, et al. [25] It works by sending several UDP probes from the source to each node in the path, and estimates the latency and throughput based on the round-trip time. When using Clink we probe each link ten times for each packet size, with an interval rate of 5 times the round trip time between packets.

Figure 6 shows the approximate positions of each node. We tried to maintain the latency of each link lower than one millisecond while maximizing the distance between the nodes. The latency of each link is shown in the picture (lat), as well as the estimated throughput (tp). A snapshot of the experiment with the final positioning can be seen in Figure 5. Note that the first hop link does not have line-of-sight, while the others have.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2017

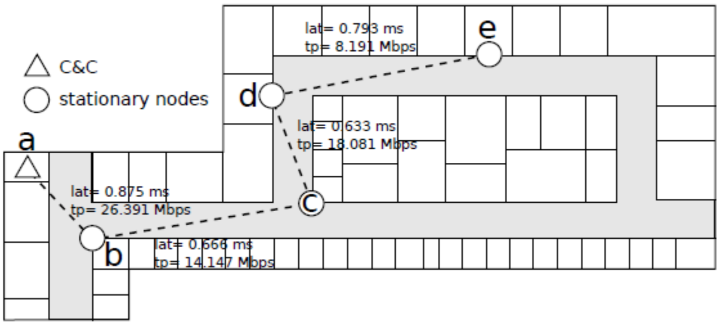

Figure 6: Approximate node positioning. Triangle is the fixed sink, circles nodes can be rearranged until satisfactory path characteristics are found.

2. Hop-by-hop degradation:

In this experiment, we want to visualize the degradation in performance when the number of hops in the end-to-end communication increases. With the nodes in the positions as in Figure 6, we measure the throughput, the number of retransmissions, jitter, and packet delivery ratio (PDR). Using the Iperf [26] tool, we obtain the throughput and retransmissions using TCP, while jitter and PDR use UDP. We average the result over 600 seconds. At each command execution, we change the source/destination pair. The sink is always the base node, while the source varies according to the desired hop count.

Figure 7 gives the results obtained. The overall trend is as expected; the throughput decreases as the number of hops increases. The number of TCP retransmissions also follows this trend. The small peak on hop number 2 can be the result of the synchronization process triggered by OLSR or uncontrolled environmental variables (people walking by, opening doors, etc.), but it does not affect the overall performance as we can see. We also see that the metrics suffer more drastically when four hops are considered. We can attribute this behaviour to the link characteristic of the last hop, where the throughput considerably drops in the last link of the path, as we can see in Figure 6.

3. Multi-flow:

In this experiment, we investigate how the network performs when multiple traffic flows from a source/destination pair exist. Using the same setup as the previous experiment, we average the performance per flow in a 600 seconds trial. We fix the source/destination as the farthest node and the sink, respectively. We vary the number of flows from 0 to 5.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2017

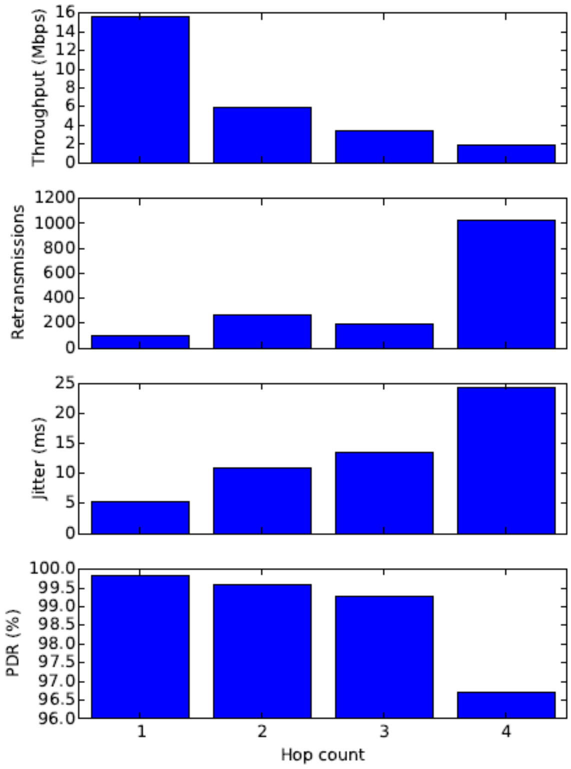

Figure 7: The effect of hop count in the end-to-end performance.

Figure 8 shows the results of the multi-flow experiment. The aggregate value is the sum of all the individual values of the flows, and the average value is the total divided by the number of flows. Using TCP, the achievable through put per flow drops as the number of flows increases. The same trend can be seen with the total number of retransmissions. Counter-intuitively, the number of retransmissions per flow decreases instead of increasing. This stable pattern happens due to the TCP congestion avoidance algorithms. It adapts the rate it pumps data into the network according to the feedback received (i.e. timeout or duplicate ack). Thus, if the throughput observed changes, it changes the rate (congestion window) it sends data, avoiding the overload of the network.

Using UDP with fixed throughput of 1 Mbps per flow, we measured the jitter and PDR. The total jitter increases as the number of flows increases, while on average the jitter per flow is stable. Here we can also observe the effect when we flood the network over its capacity. The PDR has a stable trend up to the point where we overload the network with more data than it can handle (UDP has no congestion control algorithm), which is why the peak occurs with 5 flows. In practice, this would cause all services in the network to fail or underperform.

4.2 Mobile scenario:

In this experiment, we use the setup from Figure 9, where one node is mobile (node e), and the rest have fixed positions, including the C&C (node a). It shows how the network performance behaves while the robot is moving. We collect the data for 7 minutes, which is the approximate time the robot takes to complete a full lap. The objective is to show how the network converges and adapts while the node is moving. The farthest it moves, the more the performance is affected. For this experiment we use a combined set of tools: Iperf measures throughput, congestion window size, jitter, and PDR, while trace route and ping give hop count and delay throughout time, respectively.

In Figure 10, we see the performance adaptation of the mobile system in comparison to the static setup. The benchmark used is a 4 hop communication between the sink and the farthest node e. The vertical dashed lines show approximately the moment when the network self-organizes to adapt to the new condition, where the end-to-end communication path needs one more hop to stay connected.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2017

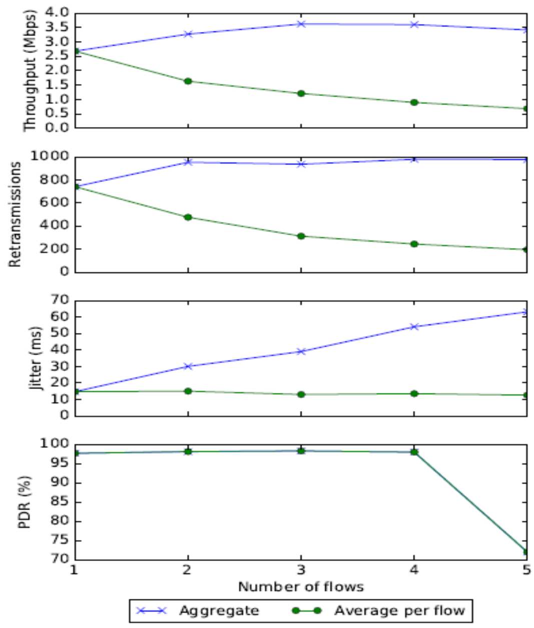

Figure 8: Number of flows comparison.

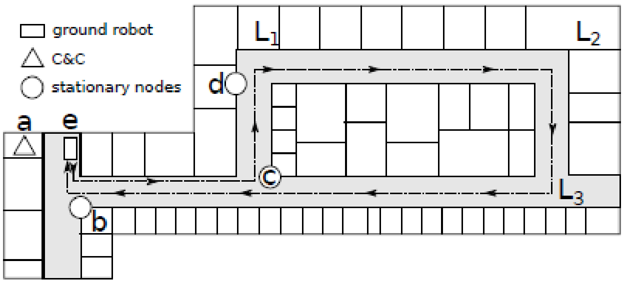

Figure 9: Multi-flow experiment with mobile node. Dashed line represents the trajectory of the robot.

As the mobile robot starts moving exploring, it is close enough to the C&C, thus communicating directly. As it moves past node b the communication continues to be direct. However, when it moves towards node c, it is not able to connect to the C&C directly anymore. The node then adjusts the communication autonomously so that it now relays the communication through b to reach the C&C in two hops. Same thing happens when the mobile robot goes past nodes c and d, adding hops to enable the end-to-end communication between e and C&C.

The throughput drops to zero as the robot turns the corner denoted by L2, where the robot leaves the communication range. Once it turns around the corner L3, it reconnects, but not immediately. Since the end-to-end communication happens on the transport layer and the lower layers have to converge before the connection is reestablished. The benchmark performance maintains an average throughout the experiment. The mobile performance oscillates around this comparison threshold.

To evaluate how the system behaves when multiple flows are present, we repeat the experiment in IV- 4.1.3 and the source node being mobile. Figure 11 shows the performance variation between each flow. We can see the resources are uniformly divided among the flows, but as the mobile node moves away from the sink, the throughput of each flow also starts to vary. Throughout the experiment no flow prevails with the highest throughput; the division of the resources do not benefit one single flow. Similarly, the number of retransmissions and the size of the congestion window (cwnd) of each flow also follow the same behavior.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2017

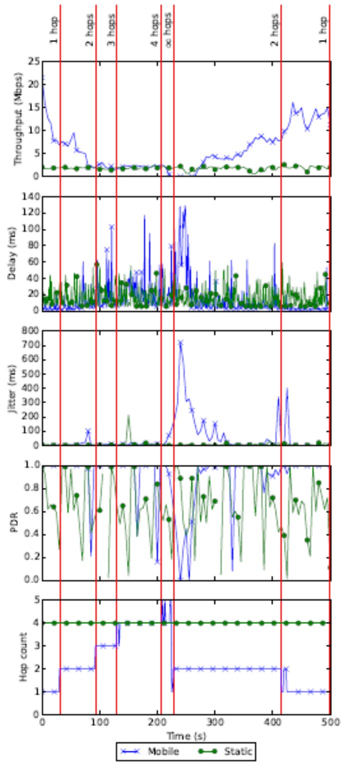

Figure10: Performance throughout the duration of the experiment.

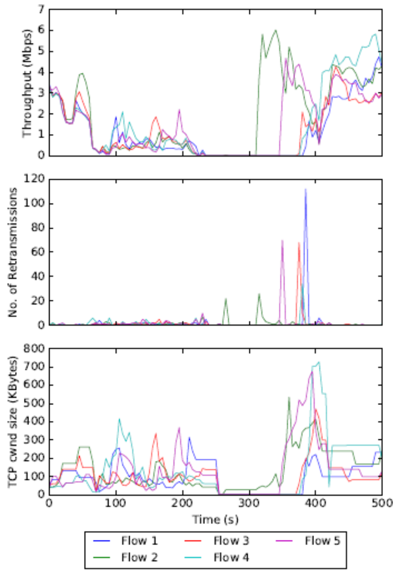

Figure 11: Multi-flow comparison with mobile node.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2017

V. CONCLUSION AND FUTURE DIRECTION

In this paper, we assembled a simple mobile mesh network testbed based on COTS products and open-source software. The system provided insights on real situations and allowed to evaluate various scenarios. We investigated the performance of the network in a real environment. The experiments showed how certain uncontrolled variables could affect the system performance. It also showed the issues with the performance when the network scales up. In the future, we plan to make the testbed autonomous to enable mission-centric experiments in critical scenarios for first responders.

VI. ACKNOWLEDGMENTS

This research is partially supported by the NSF CAREER Award #1346600, NSF Partnership for Innovation Award #1430328, and CAPES Brazil 13184-13-0.

VII. REFERENCES

1. X Dionysis, P Nikos, et al. Mobility management for Femtocells in LTE-advanced: key aspects and survey of handover decision. IEEE Communications Surveys and Tutorials 2014; 16:64-91.
2. J Minho, M Taras, et al. A survey of converging solutions for heterogeneous mobile networks. IEEE Wireless Communications 2014; 21:54-62.
3. IF Akyildiz, W Xudong, A survey on wireless mesh networks. IEEE Communications magazine 2005; 43:23-30.
4. X Xiaoshuang, J Tao, et al. Channel quality prediction based on Bayesian inference in cognitive radio networks. INFOCOM, Proceedings IEEE 2013: 1465-1473.
5. L Xingya, X Jiang, A practical self-adaptive rendezvous protocol in cognitive radio ad hoc networks. INFOCOM Proceedings IEEE 2014: 2085-2093.
6. EAA Abdulla Ahmed, F Md Zubair, et al. An optimal data collection technique for improved utility in UAS-aided networks. INFOCOM Proceedings IEEE 2014: 736-744.
7. S Mukesh, A Abdulhameed, et al. How close are we to realizing a pragmatic VANET solution? A meta-survey. Computing Surveys (CSUR) 2015; 48.
8. A Adrian, C Ciprian-Romeo, et al. Implementation of a cost-effective V2X hardware and software platform. IEEE Communications (COMM) 2016:367-370.
9. https://greatscottgadgets.com/hackrf/
10. https://www.arduino.cc.
11. https://www.raspberrypi.org.
12. https://github.com/regisin/olsrd.
13. WW Jan, S Bijorn, et al. V2X-Based traffic congestion recognition and avoidance. International Symposium on IEEE Pervasive Systems, Algorithms, and Networks 2009; 637-641.
14. E Nnanna, L Chau, et al. Towards experimental evaluation of intelligent transportation system safety and traffic efficiency. IEEE International Conference on Communications (ICC) 2015; 3757-3762.
15. T Gregor, I Colwill, et al. Mission aware topology healing for battlefield MANET. Journal of Battlefield Technology 2014; 17:7-12.
16. K Murat, P Manisa, et al. Communication network requirements for major smart grid applications in HAN, NAN and WAN. Computer Networks 2014; 67: 74-88.
17. G Florian,G Guette, et al. Comparing inter-domain routing protocol assessment tools for MANET. IEEE International Conference on Communication (ICC) 2016:1-6.
18. H Songfan, W Muqing, et al. Performance comparison of AODV and DSR in MANET test-bed based on internet of things. IEEE Vehicular Technology Conference (VTC) 2015: 1-5.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 8, August 2017

19. D Johnson, G Hancke, Comparison of two routing metrics in OLSR on a grid based mesh network. Ad Hoc Networks 2009; 7:374-387.
20. A Banerjee, J Cho, et al. Phantomnet: Research infrastructure for mobile networking, butt computing and software-defined networking. GetMobile: Mobile Computing and Communications 2015; 19:28-33.
21. M Ott, I Seskar, et al. Orbit testbed software architecture: Supporting experiments as a service. IEEE Testbeds and Research Infrastructures for the Development of Networks and Communities 2005; 136-145.
22. B Admir, B Tetsuya, et al. Experimental results of a raspberry pi based WMN testbed for different OSs in indoor environment considering LoS scenario. International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA) 2015: 207-212.
23. P Jacquet, T Clausen, Optimized link state routing protocol (OLSR). Tech. Rep.
24. https://github.com/AllenDowney/clink.
25. DB Allen, Using pathchar to estimate internet link characteristics. Proceedings of the conference on Applications, technologies, architectures and protocols for computer communication 1999; 29: 241-250.
26. http://software.es.net/iperf/