An Accessible, Open-Source, Realtime AODV Simulation in MATLAB

Stuart Miller
Department of Electrical and
Computer Engineering
Missouri University of Science & Technology
Rolla, Missouri 65409
Email: mailto:sm67c@mst.edu
Web: http://web.mst.edu/~sm67c

***Abstract*—It is often necessary to provide a base-level teaching tool when explaining concepts to new learners, or for the engineer performing initial basic prototyping. Understanding routing schemes in wireless ad-hoc networks is a central concept for any new learner. Since the subject is best taught through hands-on experience, many learners find themselves at a disadvantage when exposed to even more confusing and highly specialized tools used in wireless network simulation. Therefore, it is proposed to use the familiar and accessible environment of MATLAB to create base-level accessible, open-source, realtime ad-hoc routing scheme simulations, here targeting the ad-hoc on-demand distance vector (AODV) routing scheme.**

I. Introduction

One of the key advantages to using MATLAB as an environment is its natural support for high quality GUI displays. The suer interface is easy to understand and can be updated with simulation data in real-time. A major drawback to many preeminent network simulation tools is their reliance on written scripts and their inability to allow real time analysis and modification. MATLAB allows real-time input/output interaction and visualization. Additionally, it provides a much better debugging environment.

Perhaps most importantly, this work can provide a much more accessible intro to wireless routing schemes, as most engineers are already familiar with MATLAB functionality. Most network simulation scripting environments provide a significant barrier to enter due the their learning curve and often extensive environment setup time. MATLAB requires no setup beyond installation of the base program and is easy to understand. It follows that Stack Overflow listed MATLAB as one the least disliked languages of 2017 [1] and that 4.3% of all contributors used MATLAB as their primary language [2] (although undoubtedly more are familiar with it causally).

For the purposes of this initial work, the focus will be on the ad-hoc on-demand distance vector (AODV) routing scheme; however it is easy to see how the concepts here can be extended to additional routing methods.

II. Literature Survey

Although AODV was developed almost 15 years ago [3], it still remains highly studied in the current state of the art. The recent body of work shows much interest on the subject of AODV as a routing protocol. Several recent comparative analyses of routing protocols in MANET or VANET networks feature AODV: Ferrnato, et. al. in their protocol analysis with regards to urban vehicular scenarios [4], Abuashour and Kadoch in their study of clustered VANET scenarios [5], and Ejmaa, et. al. in their modification of AODV to create a routing scheme with less RREQ message overhead [6]. The abundance of work in this area serves as assurance that AODV still remains a central protocol to the field and will serve as a relevant target protocol for this project.

While MATLAB does already provide a comprehensive WLAN systems toolbox [7], it is solely targeted at IEEE 802.11 standards-compliant LAN-based networks and provides little-to-no support for ad-hoc or sensor networks. The selection of specified ad-hoc routing schemes and modification of node parameters is not present in the toolset provided. Unfortunately, this is far from sufficient for the goals of this endeavor.

The existing body of work does, however, show several instances of using MATLAB to test out highly specialized ad-hoc routing schemes. Gaur, et. al. show a novel routing scheme for optimizing battery life of the mica2 motes using MATLAB as a simluation environment [8]. Navya and Deepalakshmi utiltize MATLAB so show the feasibility of their proposed M-TSIMPLE routing protocal, an energy-efficient routing scheme for wireless body area networks [9]. While both of these approaches do prove the feasibility of MATLAB as a simulation environment, each is far too specialized for the generalized application needed here.

Habib, et. al. describe another approach to the usage of MATLAB in this area: to simulate path-finding algorithms in various routing schemes, and to provide a comparison among networks of varied size [10]. Their analysis focuses more on the mathematics of pathfinding and as such, is also insufficient for our purposes.

While there is significant work being produced using MATLAB as an environment, much of the groudbreaking work is being done in other simulation tools as well. Network Simulator 3 (ns-3) a popular tool for such simulations; it provides a discrete-event network simluator/emulator and scripting language based on C++ or Python code [11]. Ns-3 is targeted primarily for research and educational use. While popular, is has received significant criticism. Patel and Kmaboj criticized ns-3 harshly for it's lack of a GUI and noted it's complexity and lack of hosted work and support [12]. That is not to say it is not without significant work in the academic community. Mai, et. al. used ns-3 to simulate their version of AODV, modified for a better congestion control scheme [13].

Another noteworthy platform is NetSim [14]. Netsim Academic is the tool that most closely mirrors the intentions of this project as it is targeted at teaching/learning applications and provides adundant examples and support. Unfortunately, it is neither free nor open-source and requires a large license purchase to use. It is not without usage in the academic community though; numerous papers have cited NetSim as their primary simulation environment. Saifuddin, et. al. utilized NetSim in their analysis of alternative channel allocations for wireless networks [15]. Nayak and Sinha provided an analysis of various mobility models for routing protocols (including AODV) using NetSim [16]. Figure 1 shows an excerpt from their paper featuring their AODV simulation. A setup similar to the is the target for this MATAB implementation.

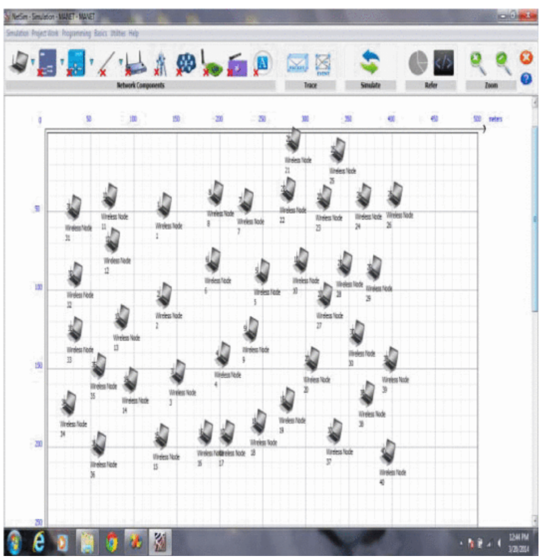

Fig. 1: Nayak and Sinha's AODV simulation window [16]

Finally, OMNeT++ is yet another discrete-event library used frequency used to build network simulators [17]. OMNeT sees quite a bit of use in the academic comnmunity as well, Chadha, et. al. used it in their simulation of energy consumption of nodes in a 3D environment [18].Thomas and Irvine investigate bandwidth usage in LTE sensor networks using OMNeT as a simulation tool [19]. Unfortunately, OmNeT falls into the same category as ns-3 in that it is overly complex and features far too steep of a learning curve to serve as a useful comparison for the objectives here.

III. Methodology

All of these simulation tools fall short in that they require a large learning curve; an engineering student or learner wishing to construct a prototype in any of these environments must first spend significant time learning the intricacies and methods of each. Furthermore, none are realtime and most provide little or no real support for visualization. MATLAB as an environment provides these features out of the box. Uluisik and Sevgi summarize this sentiment quite appropriately in their paper developing a MATLAB GUI for mapping complex functions[20]; after summarizing the existing body of work, they determined that an inherently simple and easy-to-operate MATLAB package was the best means to demonstrate the concepts they wished to teach.

MATLAB provides a single high fidelity environment that facilitates both easy plotting of data and quickly creating robust user interfaces. The AODV simulation utilizes multiple figure windows that can simultaneously display different aspects of the algorithm's progress. It will also have the capability to plot statistical data afterwards.

The AODV simulation will start by allowing a layout scheme to be chosen and a set number of nodes to be plotted on a position grid (this project will focus on only 2-dimensional positional layouts). The user can add, remove, and manipulate nodes as desired and to create unique contrived network scenarios to test edge cases Each node will then be able to ”flood” its nearest neighbors and build a local routing table. The user can then request packets be sent between two nodes and observe the path in real time.

As the intent of the project is to provide a basic level teaching and learning tool; the simulation in its current state will skip some aspect of wireless network simulation. Notably, this simulation does not implement queuing. Interference, sources of time delay, and power utilization are all removed so that the learner can focus more on the basic principles at play.

IV. Results

A. Example 1

The simplest scenario involves a node sending a transmission across the network with no prior knowledge whatsoever. Take the node layout in in figure 2. Lines show the available links to each node's nearest neighbors. A packet will be sent from node D to node G. D has no route to G so it emits a route request (RREQ), which is forwarded by each node that receives it and thus propagates throughout the network via ”flooding” (notated by the cyan links). Along the way, each node sets up reverse route entries detailing the next node, hop count, and sequence number in order to send to the source of the route request. Upon receiving the request, node G replies with a route reply (RREPL)(notated by the blue links), which the intermediate nodes know how to direct thanks to the reverse routes that were just set up. Finally, now that D has a valid route to G, data is sent (notated by the green links). Subsequent transmissions to G from D will continue to use this route and require no new overhead.

As an aside, this simulation does not visualize any RREQ messages that are not acknowledged. In reality, each node is sending out RREQs in all directions, including the direction

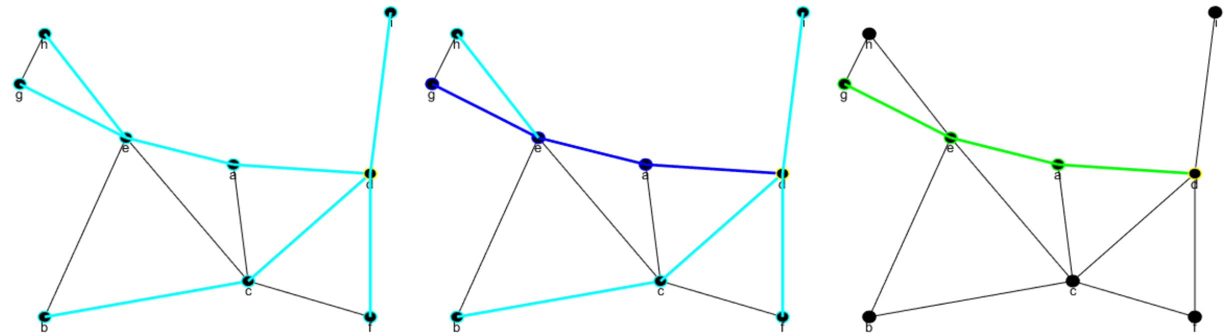

Fig. 2: Simple route request and reply, route messages
RREQ, RREPL, and data

AODV Sim - Table View

SeqNum: 1 — **Node a**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	d	1	1	1
2	g	e	2	1	2

SeqNum: 1 — **Node b**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	c	2	1	1

SeqNum: 1 — **Node c**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	d	1	1	1

SeqNum: 1 — **Node d**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	g	a	3	1	2

SeqNum: 1 — **Node e**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	a	2	1	1
2	g	g	1	1	2

SeqNum: 1 — **Node f**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	d	1	1	1

SeqNum: 1 — **Node g**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	e	3	1	1

SeqNum: 1 — **Node h**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	e	3	1	1

SeqNum: 1 — **Node i**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	d	1	1	1

Fig. 3: Simple route request and reply, node route tables

that it just received it's own RREQ from, but that node will discard it upon receipt.

Upon completion, each node's route table can be ovserved in the figure 3. Note that nodes contain a reverse route entry for D thanks to the flooding. Nodes A, D, and E are along the data path and contain routes to G.

B. Example 2

Example 2 involves utilized the previously stored route tables (figure 4. In this case, node C will try sending to node G. Since C contains no entry for G in its route table, it must flood; however, the intermediate nodes A, D, and E all do have entries for G. Instead of propagating the RREQ message onward, they respond with a RREPL. C updates its route table upon receipt of each RREPL, choosing the route with the lowest hop count to G each time. Eventually, C is able to send out to E, who the continues to send to G via its route entry established in Example 1.

Again, the route table is shown in figure 5. Note the additional entries in A, B, D, E, and F.

C. Example 3

One of the most beneficial features of AODV is its elegant handling of link failures. A link failure occurs when a pre-established route is no longer possible due to node movement or interference. AODV handles this by sending route error (RERR) messages, as shown in figure 6. In this example, node D once again tries to send to node G, but intermediate node E has moved, breaking the route. D starts off by trying to send normally, but A must reply with a RERR when it realizes it can't reach E. This message propagates back up to D and both nodes cancel out their routes to G. D must once again flood by sending out RREQ packets. C replies that it knows a route (the one established in example 2) and D then proceeds to send to G via C.

The subsequent routing table is shown in figure 7. The sequence number have increased several times. Each time the nodes notice a change in their local network topology the increase their sequence numbers. All nodes connected or disconnected from E have been incremented to 2.

While this is an exceptional method of handling single link failure or small changes in topology, it starts to become rather burdensome when multiple route failures occur. One can imagine a contrived highly mobile scenario where the topology changes in such a way as to allow every intermediate node to obtain a false route entry for the desired destination. In such a case, the source node must attempt to send and receive a reply. It then floods but receives multiple replies. Because AODV provides no method of maintaining a node's past activity, it must attempt sending again, receive an error, flood again, attempt to send again, etc. It can only cancel out one intermediate node's route destination entry at a time, potentially causing enormous overhead.

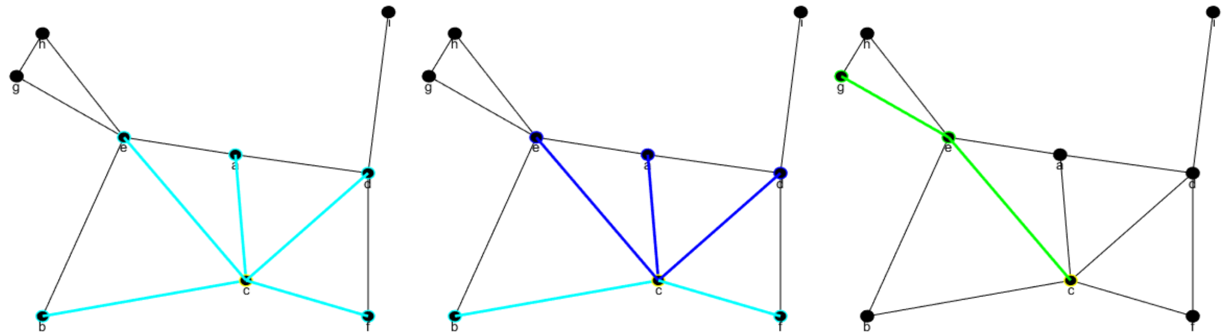

Fig. 4: Route request and reply from intermediate nodes, route messages RREQ, RREPL, and data

AODV Sim - Table View

SeqNum: 7 **Node a**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	d	1	1	1
2	g	e	2	1	3
3	c	c	1	1	1

SeqNum: 3 **Node b**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	c	2	1	1
2	c	c	1	1	1

SeqNum: 1 **Node c**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	d	1	1	1
2	g	e	2	1	2

SeqNum: 1 **Node d**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	g	a	3	1	3
2	c	c	1	1	1

SeqNum: 3 **Node e**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	a	2	1	1
2	g	g	1	1	4
3	c	c	1	1	1

SeqNum: 5 **Node f**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	d	1	1	1
2	c	c	1	1	1

SeqNum: 3 **Node g**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	e	3	1	1

SeqNum: 1 **Node h**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	e	3	1	1

SeqNum: 5 **Node i**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	d	1	1	1

Fig. 5: Route request and reply from intermediate nodes, node route tables

V. Analysis

To investigate the problem of quantifying exactly how much overhead various network topologies generate, statistics from several repeated transmissions were aggregated. In each simulation, the network had it's nodes reset to an established starting position and the route tables for all nodes were cleared. Nodes were set in motion and updated their position regularly. Each node had a pre-established step size and direction. Nodes reflected off the boundaries of the simulation area in order to ensure that they stayed relatively close to each other and did not go out of bounds. Random traffic was generated and sent through the network. The random number generator used a static seed to ensure that all variables were consistent from experiment to experiment.

To begin with, traffic was evaluated on a static network (figure 8). Here, the number of RREQ messages steadily increases until around the 50th packet, at which time most nodes have all possibly connections mapped out. The data rate climbs steadily since no transmissions fail and the RERR rate is accordingly zero.

A more realistic scenario may involve more frequent movement. Figure 9 shows overhead with movement every 50 packets and figure 10 with movement every 10 packets. The frequent movement in figure 10 finally surpasses the data rate and would likely be enough to overwhelm the controllers on most realistic nodes. In such a high mobility case, a specialized routing algorithm would likely need to be utilized as AODV's performance is lacking here.

Figure 11 shows the number of hops required for each transmission. Although the numbers are not initially as dissimilar as might be expected, it is clear that the ”movement every 10 packets” data set had consistently higher transmissions counts towards the upper end. The ”no movement” data set also had a few towards the high end as well. This represents the initial flooding, which requires all nodes in the network to send messages. This flooding is required, to some degree, at all levels and is what evens out the hop count statistics here.

VI. Future Work

Although this simulation provides an excellent base-level teaching tool, it falls short of accurate analysis in several areas. The foremost improvement that could be made would be to implement node queuing. While this would require a large burden of additional work, it would garner some interesting insights into node mechanics and provide a more realistic reflection of how ad-hoc networks truly perform.

Additionally, to enhance the usefulness of this tool, it would be beneficial to be able to compare AODV to another routing protocol like destination-sequenced distance-vector (DSDV) or dynamic source routing (DSR). Being able to compare and contrast the differences and weight the benefits of AODV would be a valuable opportunity.

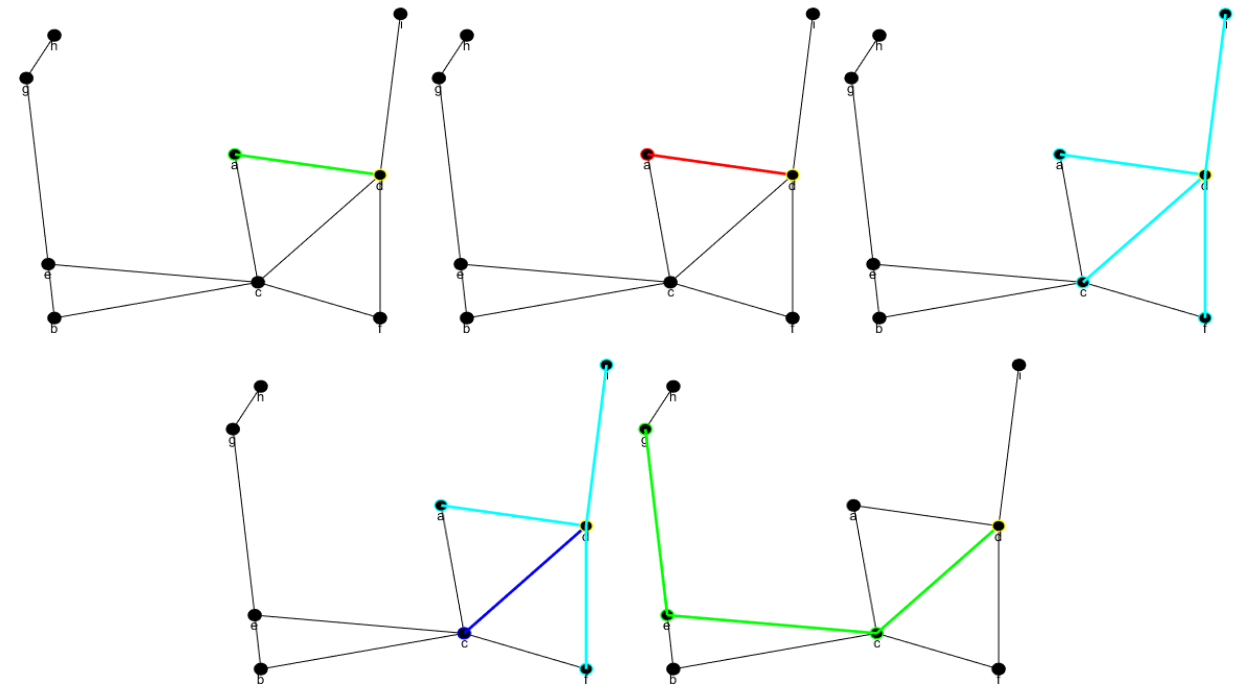

Fig. 6: Link breakage, route messages
RREQ, RREPL, and data

AODV Sim - Table View

SeqNum: 2 **Node a**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	d	1	1	2
2	c	c	1	1	1

SeqNum: 1 **Node b**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	c	2	1	1
2	c	c	1	1	1

SeqNum: 1 **Node c**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	d	1	1	1
2	g	e	2	1	3

SeqNum: 1 **Node d**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	c	c	1	1	1
2	g	c	3	1	2

SeqNum: 2 **Node e**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	a	2	1	1
2	g	g	1	1	4
3	c	c	1	1	1

SeqNum: 1 **Node f**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	d	1	1	1
2	c	c	1	1	1

SeqNum: 1 **Node g**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	e	3	1	1

SeqNum: 2 **Node h**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	e	3	1	1

SeqNum: 1 **Node i**

	dest	nextHop	hopCnt	seqNum	lifeTime
1	d	d	1	1	1

Fig. 7: Link breakage, node route tables

VII. Conclusions

The MATLAB-based ad-hoc on-demand distance vector (AODV) simulation presented above provides a meaningful method of demonstrating basic routing concepts quickly and easily, circumventing the learning curve of more specialized network simulation tools and facilitation visual learning. The MATLAB environment provides for easy inspection and expansion into additional analysis function. The examples and analysis presented serve as sufficient proof-of-concept of this endeavor to emulate AODV.

VIII. Lorem

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet,

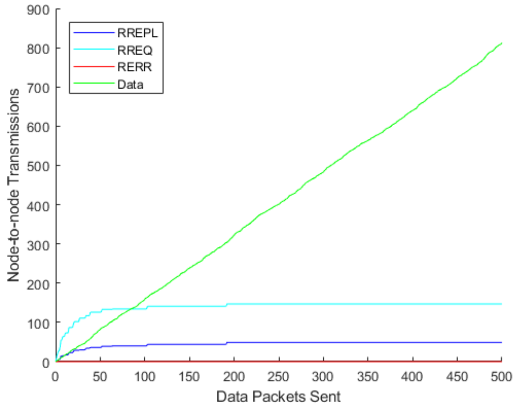

Fig. 8: Network overhead with static nodes

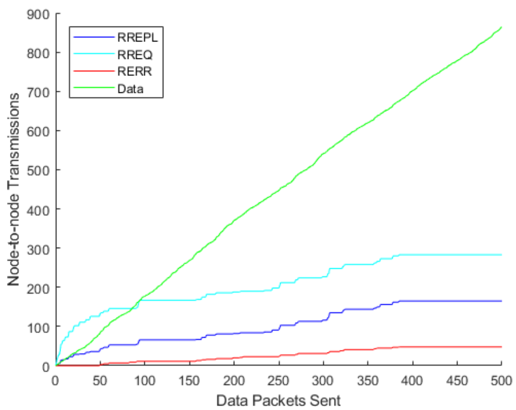

Fig. 9: Network overhead with nodes moving every 50 packets sent

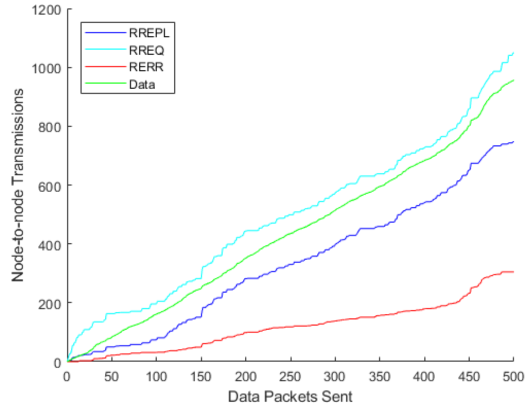

Fig. 10: Network overhead with nodes moving every 10 packets sent

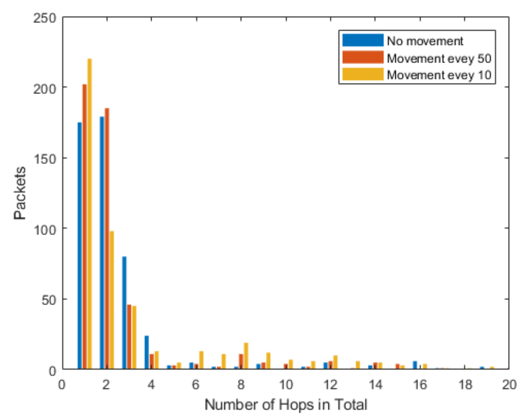

Fig. 11: Number of hops for various types of motion

tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

References

[1] D. Robinson, "What are the most disliked programming languages?" Tech. Rep., 2017. [Online]. Available: https://stackoverflow.blog/2017/10/31/disliked-programming-languages/

[2] S. Overflow, "Developer survey results 2017," Tech. Rep., October 2017. [Online]. Available: https://stackoverflow.blog/2017/10/31/disliked-programming-languages/

[3] Y. Rekhter and T. Li, "Ad hoc on-demand distance vector (aodv) routing," Internet Requests for Comments, RFC Editor, RFC 3561, July 2003. [Online]. Available: https://tools.ietf.org/html/rfc3561

[4] J. J. Ferronato and M. A. S. Trentin, "Analysis of routing protocols olsr, aodv and zrp in real urban vehicular scenario with density variation," *IEEE Latin America Transactions*, vol. 15, no. 9, pp. 1727–1734, 2017.

[5] A. Abuashour and M. Kadoch, "Performance improvement of cluster-based routing protocol in vanet," *IEEE Access*, vol. 5, pp. 15 354–15 371, 2017.

[15] K. M. Saifuddin, A. S. Ahmed, K. F. Reza, S. S. Alam, and S. Rahman, "Performance analysis of cognitive radio: Netsim viewpoint," in *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, Dec 2017, pp. 1–6.

[6] A. M. E. Ejmaa, S. Subramaniam, Z. A. Zukarnain, and Z. M. Hanapi, "Neighbor-based dynamic connectivity factor routing protocol for mobile ad hoc network," *IEEE Access*, vol. 4, pp. 8053–8064, 2016.

[7] MATLAB, "Wlan system toolbox," Tech. Rep., 2018.

[8] P. k. Gaur, B. S. Dhaliwal, and A. Seehra, "Analysis of power saving multicasting routing applications for mica2 motes-an event based matlab implementation," in *2009 International Multimedia, Signal Processing and Communication Technologies*, March 2009, pp. 109–112.

[9] V. Navya and P. Deepalakshmi, "Mobility supported threshold based stability increased throughput to sink using multihop routing protocol for link efficiency in wireless body area networks (m-tsimple)," in *2017 IEEE International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)*, March 2017, pp. 1–7.

[10] I. Habib, N. Badruddin, and M. Drieberg, "A comparison of load-distributive path routing and shortest path routing in wireless ad hoc networks," in *2014 5th International Conference on Intelligent and Advanced Systems (ICIAS)*, June 2014, pp. 1–5.

[11] ns 3, "What is ns-3?" Tech. Rep., 2018. [Online]. Available: https://www.nsnam.org/overview/what-is-ns-3/

[12] J. Patel and P. Kamboj, "Investigation of network simulation tools and comparison study: Ns3 vs ns2," *Journal of Network Communications and Emerging Technologies*, vol. 5, no. 2, pp. 137–142, December 2015.

[13] Y. Mai, F. M. Rodriguez, and N. Wang, "Cc-adov: An effective multiple paths congestion control aodv," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan 2018, pp. 1000–1004.

[14] Tetcos, "Netsim academic," Tech. Rep., 2018.

[16] P. Nayak and P. Sinha, "Analysis of random way point and random walk mobility model for reactive routing protocols for manet using netsim simulator," in *2015 3rd International Conference on Artificial Intelligence, Modelling and Simulation (AIMS)*, Dec 2015, pp. 427–432.

[17] A. Varga, "What is omnet++?" Tech. Rep., 2018.

[18] E. R. Chadha, L. Kumar, and E. J. Singh, "3 dimensional wsn: An energy efficient network," in *2017 2nd International Conference for Convergence in Technology (I2CT)*, April 2017, pp. 1169–1175.

[19] D. Thomas and J. Irvine, "Connection and resource allocation of iot sensors to cellular technology-lte," in *2015 11th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, June 2015, pp. 365–368.

[20] C. Uluisik and L. Sevgi, "A matlab-based visualization package for complex functions, and their mappings and integrals," *IEEE Antennas and Propagation Magazine*, vol. 54, no. 1, pp. 243–253, Feb 2012.