Effects of Broadcasting Over Real Ad-Hoc Networks for Distributed Mobile Robots

Noam Buckman and Jonathan P. How
Laboratory for Information & Decision Systems, Massachusetts Institute of Technology
{nbuckman,jhow}@mit.edu

Abstract—**Current hardware experiments frequently utilize slower bandwidth modes of communication for distributed robotics, relying primarily on XBee. A better understanding of the real performance over WiFi's ad-hoc mode for use in distributed multi-agent robot teams as well as algorithmic modifications to account for the issues of real decentralized communication. In this work, we test a real decentralized communication system using a high bandwidth 802.11 ad-hoc system with Raspberry Pis. We show that the communication network quickly deteriorates at high broadcast speeds, leading to worst performance as a team. In addition, while WiFi provides carrier sensing to reduce collisions, the hidden neighbor problem remains as a relevant problem for real-world systems. Finally, at high communication speeds and large network sizes, the sheer volume of messages may not be able to be processed by the communication system and further delays and missed messages can unexpectedly occur.**

I. INTRODUCTION

Communication plays a key role in multi-agent planning and estimation, serving as a critical component of the robot system. Many multirobot coordination schemes require some sort of consensus across the team such as in task allocation [1, 2, 3], consensus filtering [4, 5, 6], and coverage control [7]. In all of these planning algorithms, robots are able to estimate and plan in a distributed fashion with distributed communication. Communication is especially important in highly dynamic environments, when information can be changing very quickly and unpredictably, and thus team-wide communication is necessary to maintain a consistent team wide belief. In these dynamic scenarios it is imperative to have fast and reliable communication that still maintains the decentralized nature of most of these algorithms.

In order to implement distributed consensus algorithms on real robots, researchers utilize a few main modes of communication. A popular choice for hardware experiments is to utilize the 802.15.4 ZigBee protocol using the popular Digi XBee modules[8, 9, 10]. ZigBee provides mesh management and thus commonly used in multiagent research, however, its limitations include small packet sizes and the necessity for central coordinators – and as such – is not fully decentralized. In addition, Time Division Medium Access (TDMA) is commonly used to coordinate communication and avoid contention [7]. However, TDMA has its own limitations including initial synchronization and inefficiency for dynamic teams where robots may enter or exit. Thus for static networks and low-bandwidth message passing such as small sensor values, ZigBee may be appropriate, however for higher bandwidth requirements, such as sending over full maps of the environment or entire belief spaces, other technologies are needed.

Frequently, algorithms are tested via simulation that imposes distance or neighbor limited communication to study distributed algorithms [11, 1]. This includes using 802.11 devices (i.e. WiFi) in its centralized infrastructure mode while restricting message passing only between neighbors. While this captures the multi-hop effects of decentralized communication in large multirobot teams, real ad-hoc networks are known to have poor performance as compared to infrastructure, contention free modes [12, 13]. As such, there is a need to further explore the real performance of 802.11's ad-hoc mode and its impact on teamwide communication and consensus.

It is worthwhile to note that the consensus and robotics community has dealt with unreliable communication from the perspective of limited neighborhood of communication, limited connectivity, and random links [14, 15, 16, 17]. However, the convergence analysis of most of these systems deal with asymptotic convergence and do not completely translate to real ad-hoc networks.

For example, in broadcast protocols that is of interest is that (a) transmission success/confirmation is not practical, as the receiver is not necessarily known (b) agents can not discriminate between agents. Agents communicate with all its neighbors in range and cannot discriminate from a channel usage perspective between neighbors. This is both an advantage and disadvantage. On one hand, it is a free operation to communicate with all of its neighbors. On the other hand, there will always be a cost for communicating which is that neighbors will not be able to receive or send during that duration, and possible reception reliability will be diminished.

II. 802.11 AD-HOC WIRELESS COMMUNICATION

802.11 (or WiFi) provides two modes of communication: infrastructure and ad-hoc. In the former, a personal coordination function is used that allows for a central access points/routers to coordinate communication for higher reliability and throughput. In ad-hoc, only the distributed coordination function (DCF) is used to coordinate access to the shared medium. In 802.11 ad-hoc mode, the DCF is called Carrier Sense Medium Access (CSMA), whose main feature of interest is carrier sensing, in which it first senses the medium for

Fig. 1. Raspberry Pi Placement in 6th Floor Building 32

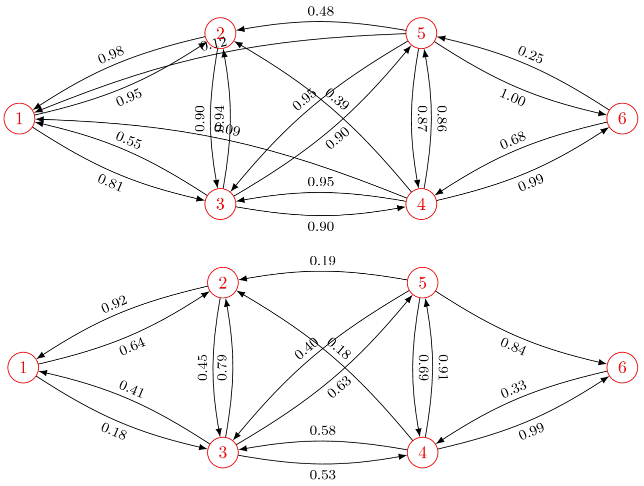

Fig. 2. Network Topology for 1 Hz and 50 Hz broadcast rates. Edges show probability of message's successful transmission.

use before transmitting, to minimize the collisions. The first effect of this is possible delays in transmission when there is high use of the channel. A naive broadcast approach across a large mobile team may flood that network with messages that will cause important information to be delayed, waiting for a free medium. While CSMA mitigates collisions with its immediate neighbors, a second unsolved issue is the hidden node problem. In the hidden node problem, lack of awareness of two-hop neighbors leads to believing the channel is free, and thus transmitting on a channel that is being used. While handshaking protocols can be implemented, including the RTS/CTS handshaking option in 802.11s DCF, in a settings with highly dynamic topologies, RTS/CTS handshaking can prove to more burdensome than the hidden node problem itself. While real experimentation on ad-hoc WiFi has been done on a few nodes as in Anastasi et al. [12], a larger network-wide analysis is lacking to understand the effect on the team as a whole.

In this work, we explore the performance of a real ad-hoc network using 802.11 WiFi adapters that are widely commercially available. We are interested in understanding the reliability and delays associated with using 802.11 ad-hoc's mode for broadcasting information across a team of agents. The goal of this paper is to better understand effects of hidden neighbors, weak link signals, and limited buffer sizes on the team's ability to transmit information.

III. DECENTRALIZED BROADCAST EXPERIMENTS

As shown in 1, a group of six Raspberry Pi 3s are places across MITs Stata Building such that it forms a non-trivial network of agents that require data fusion across the network. Each Raspberry Pi is equipped with an onboard 802.11n Broadcom chip that is used to monitor the experiments while an external Canakit 802.11n WiFi adapter is used for the ad-hoc mesh communication.

In the first set of experiments, we vary the communication rate of the Raspberry Pis to study its effect on the network as a whole. In addition, we explore the networks ability to pass information across the entire diameter of the network. In this case, we have a source node (Node 1) which passes information across the network to (Node 6). Since this is a purely distributed network, agents are not a priori aware of the source and sink nodes, and must run identical communication schemes.

The key quality of service metrics used to measure these networks will be the reliability, defined as the probability that any given transmitted packet is received between agents and the probability that some information or message is received by the farthest agent (Agent 6). In addition, we are interested in the average delays in packet transmission/reception between any two agents as well as across the entire network.

A. *The Issue of Rebroadcasting*

At any given external information rate, team-wide communication strategies may have the freedom to rebroadcast information to ensure higher probability of reception from its neighbors. At lower information rates (e.g leq 1 Hz), agents may be able to resend messages 10-100 times without reaching physical limitations of data transmission on the Wifi chip. The system designer must decide the trade-off however between increased reliability due to rebroadcasting and network degradation due to flooding. In addition, with unknown and dynamic networks, it will not be clear a priori the effect of rebroadcasts on the actual network. For example, baseline link levels and network degrees size may be unknown, and by increasing the broadcast rate too much, we will inadvertently reach the network's capacity and begin deteriorating the network.

IV. RESULTS

In the first set of results, each agent broadcasts messages across the network at the same broadcast speed. Figure 2 shows the topology of the network where graph edges represent the probability that a transmitted packet is successfully received. At a broadcast speed of 50 Hz, the topology changes completely, with entire links removed from the graph. In addition, the communication between two nodes is not

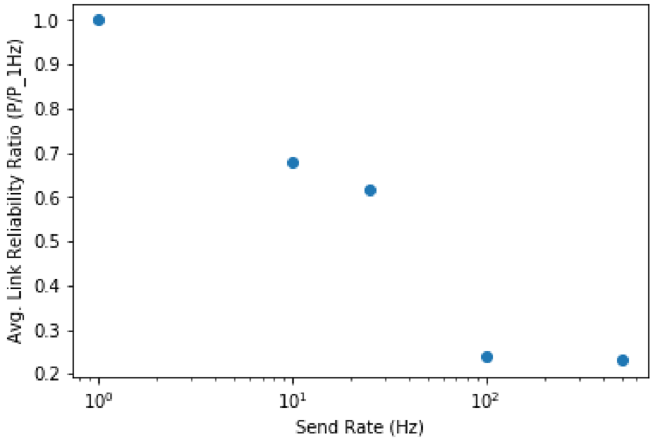

Fig. 3. Average degradation of links as ratio of link transmission probability on baseline transmission probability (at 1 Hz broadcast rate)

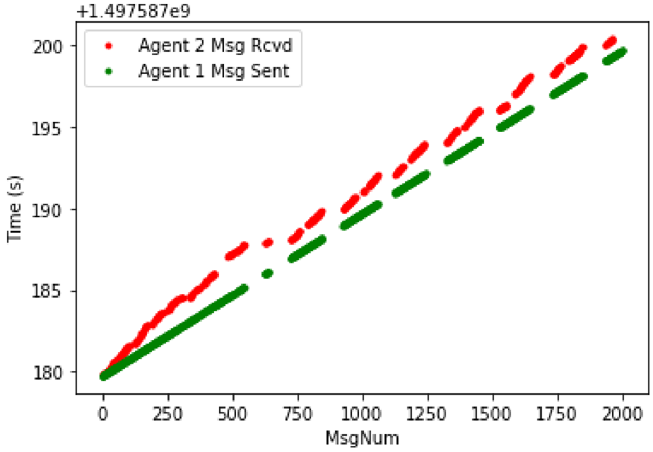

Fig. 4. Transmitted and received messages from Agent 1 to Agent 2 at 100 Hz broadcast rate (≈ 800 kbps). As more messages are sent, delays increase due to arrival rates being greater departure rates. Intermittent dropping of messages due to buffer clearing.

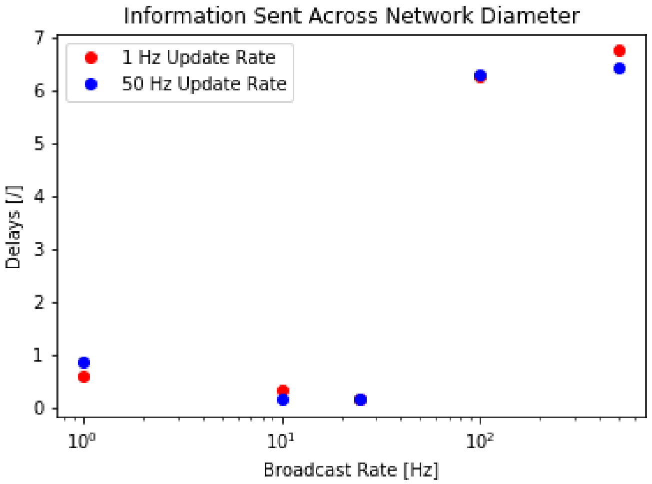

Fig. 5. Information delay across the network decreases as broadcast rates increases, however, a phase transition occurs at 100 Hz

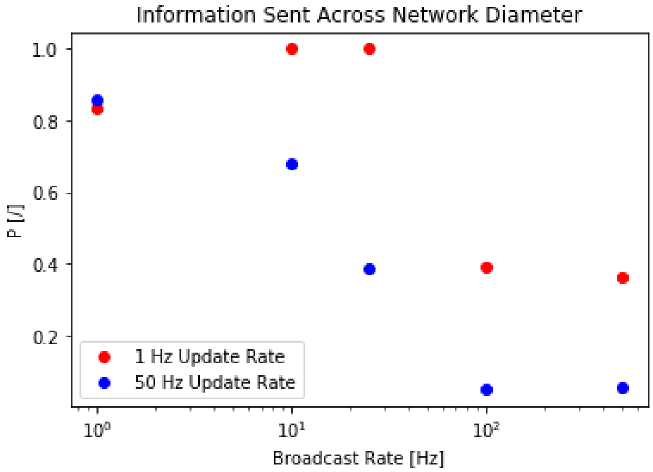

Fig. 6. Probability of message transmission across the network for various information update rates and agent broadcast rates.

symmetric, presumably due to the varying in-degree of each agent. In Figure 3, each edge is compared to a baseline graph at 1 Hz and the mean is taken to obtain a metric of relative communication degradation at different broadcast rates. As broadcast rates increase, edge probabilities degrade, achieving up to 80% worse transmission success rates than at a baseline of 1 Hz broadcast rate.

Figure 4 shows the message send times by Agent 1 and receive times for Agent 2 while broadcasting at 100 Hz. At the beginning of the experiment, there is an increase of delay times (the vertical difference between points) due to higher arrival rates than departure rates on their internal 802.11 queues. While the Raspberry Pi is checking its queue at a higher rate than anyone is sending, this limitation arises from the 802.11 chip itself, which must physically receive and process the data. Due to this overflowing of the incoming queue, there are intermittent outages and clearing of the buffers, during which entire sequences of messages are lost by Agent 2. This presents a real limitation on higher levels of broadcasting especially in large teams when multiple agents may be sending messages to an agent.

Finally, the Figures 5 and 6 explore the teams ability to transmit information across the team. In this case, Agent 1 is trying to communicate across the network (to Agent 6), some external information that is changing at some update rate. If Agent 1 is broadcasting at a rate higher than the information's update rate, Agent 1 can rebroadcast the information in subsequent transmissions. These rebroadcasts would be advantageous in increasing the probability that any one of the messages is received by Agent 6. However, the increase in broadcasts may also destroy the network, adversarial affecting the teams ability to communicate, especially when every agent maintains that same broadcast speed.

We see in Figure 5, that the delay in information reception initially decreases as agents communicate faster. However, at 100 Hz the average delay increases due to the overflow of the buffer. In Figure 6, the probability of the information getting across the network increases when the update rate is low (at 1 Hz) and thus the agent is able to rebroadcast a lot. However, for both systems, as the broadcast rate increases to much higher speeds, the system can not handle the amount of messages being sent, and probability of information being successfully received by Agent 6 goes down to less than 0.2.

V. CONCLUSION

While team-wide communication is a necessary component to effective multi-agent coordination and planning, unrealistic modeling of the network itself may prove detrimental to algorithm design. In experiments, the ad-hoc network proves to be unreliable at high broadcast speeds due to contention and overflowing of receiving buffers. Increased rebroadcasting of messages provide extra reliability that important information is received the entire teams, however, network performance is not independent of broadcast speeds and must be considered when designing real distributed robot systems.

ACKNOWLEDGMENTS

Funding provided by Lockheed Martin Corporation.

REFERENCES

[1] Han-Lim Choi, Luc Brunet, and Jonathan P How. Consensus-based decentralized auctions for robust task allocation. *IEEE transactions on robotics*, 25(4):912–926, 2009.

[2] Lingzhi Luo, Nilanjan Chakraborty, and Katia Sycara. Distributed algorithms for multirobot task assignment with task deadline constraints. *IEEE Transactions on Automation Science and Engineering*, 12(3):876–888, 2015.

[3] Stefano Giordani, Marin Lujak, and Francesco Martinelli. A distributed algorithm for the multi-robot task allocation problem. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 721–730. Springer, 2010.

[4] Reza Olfati-Saber and Jeff S Shamma. Consensus filters for sensor networks and distributed sensor fusion. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 6698–6703. IEEE, 2005.

[5] Brian J Julian, Michael Angermann, Mac Schwager, and Daniela Rus. Distributed robotic sensor networks: An information-theoretic approach. *The International Journal of Robotics Research*, 31(10):1134–1154, 2012.

[6] Saptarshi Bandyopadhyay and Soon-Jo Chung. Distributed estimation using bayesian consensus filtering. In *American Control Conference (ACC), 2014*, pages 634–641. IEEE, 2014.

[7] Mac Schwager, Brian J Julian, Michael Angermann, and Daniela Rus. Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, 99(9):1541–1561, 2011.

[8] James Stephan, Jonathan Fink, Vijay Kumar, and Alejandro Ribeiro. Hybrid architecture for communication-aware multi-robot systems. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 5269–5276. IEEE, 2016.

[9] Stephanie Gil, Mac Schwager, Brian J Julian, and Daniela Rus. Optimizing communication in air-ground robot networks using decentralized control. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 1964–1971. IEEE, 2010.

[10] Sameera Ponda, Luke Johnson, Han-Lim Choi, and Jonathan How. Ensuring network connectivity for decentralized planning in dynamic environments. In *Infotech@ Aerospace 2011*, page 1462. 2011.

[11] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[12] Giuseppe Anastasi, Eleonora Borgia, Marco Conti, and Enrico Gregori. Ieee 802.11 ad hoc networks: performance measurements. In *Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on*, pages 758–763. IEEE, 2003.

[13] Carlo Vallati, Victor Omwando, and Prasant Mohapatra. Experimental work versus simulation in the study of mobile ad hoc networks. *Mobile Ad Hoc Networking: Cutting Edge Directions, Second Edition*, pages 191–238, 2013.

[14] Stacy Patterson, Bassam Bamieh, and Amr El Abbadi. Distributed average consensus with stochastic communication failures. In *Decision and Control, 2007 46th IEEE Conference on*, pages 4215–4220. IEEE, 2007.

[15] Alireza Tahbaz-Salehi and Ali Jadbabaie. A necessary and sufficient condition for consensus over random networks. *IEEE Transactions on Automatic Control*, 53(3): 791–795, 2008.

[16] Tuncer Can Aysal, Mehmet Ercan Yildiz, Anand D Sarwate, and Anna Scaglione. Broadcast gossip algorithms for consensus. *IEEE Transactions on Signal processing*, 57(7):2748–2761, 2009.

[17] Alex Olshevsky and John N Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM Journal on Control and Optimization*, 48(1):33–55, 2009.