Evaluating Linear Network Coding for Resource Constrained Wireless Mesh Networks

Chris Henk
cahenk@smu.edu

Eric Smith
eric@smu.edu

Michael Gilbert
mgilbert@smu.edu

David Kim
dykim@smu.edu

Matthew Lee
matthewlee@smu.edu

ABSTRACT

In this paper, we evaluate linear coding in IEEE 802.15.4 based wireless mesh networks. Linear network coding has been shown to improve throughput and reduce the need for routing on nodes with sufficient computational resources. However, traditional direct network coding requires significant processing and storage that is not available on typical 802.15.4 nodes. We evaluate linear network coding on multiple network topologies including mesh, grid, and star. We test scenarios with simulated data for environmental sensors sending a combination of unicast and multicast messages. Applying network coding resulted in an increase of over 262% in average effective throughput and an 86% drop in effective load. We conclude that linear network coding is achievable on resource constrained nodes operating with the 802.15.4 protocol and that the resulting performance gains are substantial enough to warrant using network coding.

1. INTRODUCTION

The growing number of embedded, internet-enabled devices, together known as the Internet of Things (IoT), promises to stress existing Personal Area Networks (PANs) and has the potential to replace the star topology with the mesh topology as the primary PAN topology. New challenges presented by the IOT are dependent on careful analysis and, to some extent, retooling of our existing PAN protocols. In particular, the IEEE 802.15.4 protocol will need to support the growing networks sizes and transmission rates brought on by the IoT.

IoT devices form wireless mesh networks to allow for a distributed system that is resistant to faults and allows nodes to use lower power radio transmitters. This in turn improves performance and lowers cost. Contrast this with typical PANs, where we tend to find many nodes communicating with a centralized node arranged in a star topology e.g., a smartwatch, printer, and wireless speakers connected to a smartphone over Bluetooth. In this scenario, nodes are communicating directly from a single source to a single destination whose payloads are either short messages sent periodically (e.g. polling) or regular, larger streams of data (e.g. an audio stream). No routing is required, and scheduling is simplified by the low bandwidth utilization of sporadic messages and the predictability of longer ones. Mesh networks, by nature of their size and arrangement, will require routing. There may be a distributed set of controllers or individual sensors that may want to collaborate, such that the majority of nodes are no longer communicating with a single node. Nodes may have multicast traffic in addition to unicast. Clearly, there is a need for improved network performance.

It is well established that *Linear Network Coding* (LNC) increases throughput while simultaneously reducing the need for routing [3]. The ideal implementation of network coding varies for different network topologies and remains an open topic, in particular implementing network coding in a way the leverages the broadcast nature of wireless is a subject of ongoing research. Apart from its proven efficacy in the general case, network coding offers additional opportunities for improved performance as a result of the wireless mesh topology found in the IoT [5].

While network coding improves network performance, it is computationally complex. Adding network coding to any system will increase the resource requirements by a non-negligible amount. While this may be only of mild concern for more expensive devices with access to power, the proportional impact to low cost devices with limited or nonexistent power supply is significant enough to make existing implementations of LNC unfeasible [4].

Motivated by these considerations, our research measures the efficacy of applying LNC to the IEEE 802.15.4 protocol. In general, proposed solutions to the network coding problem seek to maximize performance and only aim to minimize resource utilization by minimizing waste. Typically they are not constructed with resource constrained nodes in mind [4]. For example, the maximum size of an 802.15.4 packet is a mere 127 bytes, which severely limits the ability of packets to be combined through coding and limits the use of protocol headers to orchestrate coding and/or routing [2].

The remainder of this paper is organized as follows. We begin with a discussion of linear network coding and present several implementations. Next we will introduce the IEEE 802.15.4 protocol and highlight the constraints relevant to our investigation. The description of IEEE 802.15.4 is followed by an example of implementing an 802.15.4 mesh network, using Zigbee as a real world example. With the network medium access control and topology introduced, we discuss primary routing as it relates to network coding. The introductory information finishes with a description of our linear network coding implementation for an 802.15.4 packet. Next, our experiment parameters and simulations are introduced along with the results they produce. We conclude with our evaluation of the efficacy of linear network coding in resource constrained wireless mesh networks.

2. LINEAR NETWORK CODING

Network coding was first introduced by Ahlswede et al. [1]

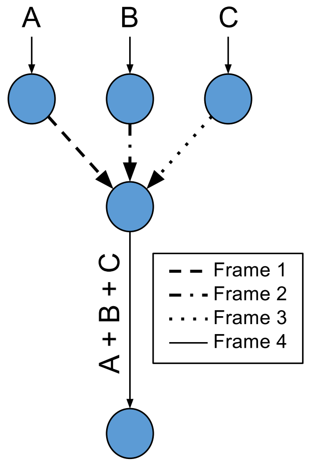


Figure 1: Wireless Network Coding Using Delays

as a technique to reduce the bandwidth utilized by multicast routing. The research proposes a class of problems denoted "Network Information Flow", which are a subset of multiterminal source coding problems and observes four stipulations. These are as follows: rate-distortion is ignored, sources are mutually independent, and the network topology and requirements for message reconstruction are arbitrary. The primary set of codes used by Ahlswede et al. [1] are alpha codes, a class of block codes, but they acknowledge that more generalized codes could provide superior performance. Specifically, they note that convolutional codes could reduce the memory requirements of the nodes and significantly decrease end-to-end decoding delay. Additionally, the results in the appendix of the paper show that probabilistic codes do not produce performance improvements.

The application of network coding to wireless mesh networks is primarily addressed by the COPE architecture introduce by Katti et al. [3]. COPE creates a forwarding architecture that improves the throughput of wireless mesh networks. This architecture utilizes network coding and exploits the broadcast nature of wireless communications. It functions between separate sessions on any topology and can handle bursty traffic. COPE utilizes several techniques in order to achieve optimal throughput. Nodes store all packets they receive, even if they are not the intended recipient. To maximize the effectiveness of this technique, each node attaches a *reception report*, which outlines all data packets that the sender has overheard, to packets it is sending, or if a node has no packets to send, it broadcasts the reports periodically to neighboring nodes. Additionally, nodes opportunistically leverage LNC by combining all packets that the next hop is capable of decoding. Finally, each node will guess what packets its neighbors have overheard in times of network congestion using a probabilistic calculation of the data that each neighboring node has received. COPE does not delay packets to wait for coding opportunities. That is, nodes send queued packets the at the first opportunity. Additionally, COPE adds its own header to a packet that will either be placed between the routing and MAC headers or between the MAC and IP headers of the original packet.

Katti et al. [3] highlight that there are multiple limitations in implementing COPE. For instance, nodes must have the memory capacity to store overheard packets and have an omni-directional antenna to optimally exploit wireless broadcast. The method also assumes that nodes have access to power and therefore does not optimize for energy usage. If the power and memory issues are addressed, COPE has the potential to serve as the basis for another architecture that addresses these issues that could serve as a solution to implementing network coding on resource constrained wireless mesh networks.

The research outlined by Krigslund et al. [4] introduces an improvement of the COPE protocol, the CORE protocol, by combining intra-session network coding and intersession network coding. CORE uses random linear network coding (RLNC) on unicast traffic without coordinating data streams. It exploits RLNC, which is typically used to provide resiliency to packet loss, to enhance inter-session coding regions. RLNC is implemented by XORing only data within RLNC packets and leaving the coding coefficients unmodified. This strategy requires all nodes to store transmissions. The protocol employs partial decoding (decoding without the purpose of extracting a packet) to recover the linear combinations from a single data stream. Nodes identify opportunities for inter-session coding where multiple unicast flows intersect and notify other nodes when they have enough information to decode a particular packet. The notifications inform other nodes not to relay information related to that packet in order to conserve energy. CORE superior to COPE in resource-constrained wireless mesh networks because this technique facilitates low power transfers with the added bonus of improving error resilience.

The research presented by Sengupta et al. [5] provides a framework to analyze network coding approaches for improving throughput on wireless multihop networks. They present a method for quantifying network coding benefits in the presence of multiple simultaneous unicast sessions. Also, this methodology estimates the throughput increase achieved by network coding on any given network topology. It proposes the idea of routing that is both coding-aware and interference aware in order to create an optimal blend of increasing coding opportunities and decreasing wireless interference. Their analysis found that in certain scenarios these routing approaches produce gains of 40% over coding-oblivious routing. They show that their model and methodology can scale to other objective functions, specifically noting the minimization of bandwidth usage in wireless sensor networks.

3. RESEARCH APPROACH

3.1 Routing

In terms of IEEE 802.15.4 network coding, primary routing within low powered networks can see large improvements through packet modifications or implementing addition motives while packets are being routed through a network. Some which include, XORing packets or switching to constant traffic. Such implementations require minimal changes to a topology and rather linear network coding can prove to be quite computationally cost efficient.

Multicast destination packets can achieve upperbound more easily than unicast since all nodes are listening for their RTS/CTS. Regardless if the wrong RTS/CTS is received, filters are searching for high priority and low priority packets. Packet priority systematically affects primary routing within a topology. High priority packets are immediately routed, causing low priority packets to be put on hold. As a result, Gaussian Elimination can take advantage by naturally combining low priority smalls packets stored within the router's queue. In other words, a bottleneck link between two nodes can be change to an efficient time-shared session – keeping the frame slotting and hops to a minimal. Intermittently, implementation of Random Linear Network Coding is not required due to the embedding of id in the header to identify which predetermined set of constants to use this would reduce the effectiveness of linear network coding considering the fact the packet size is small. In terms of using Gaussian Jordan Elimination on packets, Gaussian elimination's time complexity is lower and requires less computation, which again is better suited for our low powered network. By these forms, primary routing modification of the node must be changed to constantly monitor the channel effectively allowing network coding to take place. CSMA/CA and channel collision checks can enable nodes to naturally combine packets. Ultimately, the process of a node's routing technique can be combined with linear network coding this allowing improvements to a low powered topology such as a sensor network.

Due to the maximum 127 byte size of an 802.15.4 packet, this can limit complex algorithms to improve packet routing. Thus network coding within 802.15.4 can be focused on minimizing extraneous header information to provide extra storage for data such as temperature. An approach to this is by modifying the 802.15.4 payload by storing only the destination MAC addresses – to provide extra bytes of information to transmit. Removing the header will provide four extra bytes. This essentially will turn the nodes into broadcast nodes. The drawback to the destination-focus network will introduce decreased reliability and potential security problems. If the network topology spans multiple hops for the packets to reach their destination, this idea is not well suited. Small star and mesh networks can benefit from this possible packet payload setup. Aside that, reducing payload byte size transmission will improve power consumption especially considering traffic is constant. The saved power can be re-used for complex algorithms such a Galois to improve packet throughput or improving security by implementing a simple structure like the butterfly diagram. Additions of complexity can create sparser network topologies. In turn, testing different layouts of mesh networks, effectively providing data for the network's throughput. This will be vastly important to providing optimal topologies that are most cost effective. In other words, packet structure changes will help maximize node distance and optimal throughput through the least possible nodes to create an inexpensive, yet effective network.

However, there are constraints involved because of the additional overhead introduced when both the packet and finite field constants are sent. Because wireless sensor networks have power constraints and limited packet size, using linear network coding becomes trickier. For example, in the IEEE 802.15.4 protocol, the PHY layer is limited to about 127 bytes of data per packet compared to the 1500 bytes of IEEE 802.3. This is important to consider when implementing random linear network coding because of added overhead of the finite field constants. Furthermore, Galois arithmetic can be computationally expensive. As a result, IEEE 802.15.4 requires a different method to do these calculation, such as introducing a central device that can handle the calculations, keeping in mind any additional costs any other methods may incur. Lastly, wireless sensor networks calls for a low power system so that it can last for an extended period of time. It is also important to note the power consumption something as computationally heavy as Galois Theory. Certain chips on the ZigBee platform are designed to last in the upwards of 20 years. Doubling power consumption can mean years taken off a device's life. Even though linear network coding could still function if it took a lot of power, adhering to the low power principle of IEEE 802.15.4 is necessary and important.

Figure 2: MAC Addressing Only

Figure 3: With Network Addressing

3.2 Modified 802.15.4 Packet

Our implementation of LNC takes advantage of the relatively large, 20 byte address field and our choice of field constants. The hardware MAC addresses in 802.15.4 are 8 bytes each for the sender and the receiver, leaving 4 additional bytes for addressing and 102 bytes of payload. Two of the remaining bytes in the address field are used as the ”network mode” header. It contains the packet's field constants, priority level, and, optionally, additional routing flags. Our field constants are Galois Field constants of the form $GF(2^m)$, where $m = \mathbb{N}\epsilon[2, 5]$. The value of m can be set independently on each packet to scale with the number of packets that the node wishes to encode. Since the field constants are fixed values that can be computed or looked up knowing only the value of m, there is no reason to consume valuable space sending the constants themselves. Instead, we send only the value m encoded as a string of three bits that are offset by -2 e.g., $m = 2$ is encoded 00, $m = 3$ is encoded 01, and so on. This approach eliminates the issue of field constant overhead by reducing it to a fixed 2 bit overhead.

The packet priority is encoded as 6 bits and represents the maximum number of frame slots a packet may be delayed when waiting for additional packets to combine it with. The delays these bits correspond to is implementation specific and should be set using a control frame. The remaining 8 bits in the ”network mode” header is intended to exchange various implementation specific routing flags that can tune the network layer to its specific usage. Finally, our methodology allows 2 bytes each to optionally be used to assign network layer addressing. Sensor networks are unlikely to have more than 2^16 nodes in most cases. If this were to become problematic, a formal standard for our methodology could assign some of the extra bits in the ”network mode” header to address concerns, however, our simulations were run on networks where this was not an issue. Depending on the addressing mode chosen, our methodology provides routing and LNC while retaining 100-104 bytes of the original 127 bytes for data payload. Figure 2 provides a visual breakdown of our packet structure with network addressing disabled and figure 3 provides a visual breakdown of our packet with network addressing enabled.

4. EXPERIMENT

4.1 Tested Topologies

Our experiment tests star, grid, and mesh topologies. Our simulations were designed with residential and commercial office building environments in mind. Residential environments can range from one room to the entire property, while commercial buildings can range from a small office space to a small office complex. When determining which topologies to use in our simulations, we assumed that each pair of nodes connected by an edge were at most 10 meters apart, as 10m is the average sensor range in 802.15.4. With this constraint in mind, we can construct each topology to apply to a residential case and/or a commercial case. We use multiple topologies to test how various arrangements of low powered 802.15.4 nodes impacts the communication between them. Furthermore, we can make our own adjustments to the 802.15.4 protocol to see if we can make improvements based on the topologies defined below.

As sensor networks become increasingly complex, the need for mesh networks also increases to address the nodes' need to communicate with one another in more complex manners than those provided by star or even grid. However, there is still value in testing the star for more simplistic, centralized control sensor networks and grid for scenarios that map well to a grid structure (e.g. a sensor network for stoplights in a city). By comparing these results we accumulate from running the simulation using mesh networks, we can identify the performance characteristics of mesh networks focused around the actual communication and throughput of packets through an edge on a network graph. This is excluding factors such as cost, complexity of initial setup, etc.

The star topology is one of the more abundant topologies in use today, especially in the personal electronics sector. Though it is difficult to say that it will still be the most often used topology in a sensor node environment, this topology is important to test with the intention of leveraging the already established infrastructure. This lessens the need to change the network infrastructure to something completely new. Consequently, we ran tests and studied the results of star topologies ranging from small to large. To reduce error in our results, we conducted one thousand randomized trials for each set of test parameters by randomly generating packet addresses and creations times, since the arrangement of nodes itself is constant with a fixed number of nodes arranged in a star topology.

The grid network serves as something of a mix between the predictable structure of the star network and the random configuration of the mesh network with each node having multiple connections to other nodes. The purpose of this test is to serve something of an intermediary step between star and mesh that helps us see a measurable progression between predictable to unpredictable structures. Just as with the star topology simulations, we reduced error in our results by conducting one thousand randomized trials for each set of test parameters. Again, reach trial involved randomly generating packet addresses and creations times, rather than the network layout, since the arrangement of nodes itself is constant with a fixed number of nodes arranged in a grid topology.

Finally, the mesh network was tested to determine what throughput improvements, if any, could be realized when applying LNC to the topology that most sensor networks will be using. This topology is the one we put the most focus on because of its applications and potential widespread use in a sensor network. Drastic improvements in the throughput of a network that employs linear network coding justify its implementation in real world sensor network applications as an effective method to both minimize power consumption and therefore cost. In our simulations, the topology generation script generated one thousand different topologies with random edges between nodes, using the same packet file to control the number of variables changed.

4.2 802.15.4 and Tested Parameters

Because this research targets the IEEE 802.15.4, many of the parameters like packet size and packet encoding will be based off of the 802.15.4 standard. As stated before, we use the assumption that 802.15.14 works off of a base range of about 10 meters. Any difference in sensor range will be ignored for the scope of this project. Dijkstra's shortest path algorithm run at each node will suffice as a method of creating routing tables for each.

The simulation also tests the ability to transmit unicast and multicast destination packets from a single source, improvement with encoding a packet into multiple in regards to packets and packet structure. Test parameters involving the network structure itself include the performance drop a network would endure with a change in size and/or structure measured by the number of nodes, tested under each of the above mentioned topologies. Furthermore, the mesh topology would have a range of different node organizations and edges. The simulations themselves test if 802.15.4's protocol allows for efficient packet communication. If this communication is possible, what parameters can be adjusted to improve communication between nodes? If not, which parameters can we revise to make talking possible? Answering these questions is the overarching goal of our research.

Another parameter provided was the notion of packet priority where a certain packet has a high priority that will be sent first over a low priority packet that may have arrived at a node earlier. With packet priority, a network can push emergency information faster than a 'fair' network such as in the situation where a patient in a hospital has a sudden, dangerous dip in blood pressure and every second of response can be significant. Packet priority was chosen as a random ratio ranging from 1% to 99% (i.e. 85% of packets are high priority).

Along with the different number of destinations for packets (unicast versus multicast), the simulation ran different ranges of values to test linear network coding for different network configurations. Each of the three topologies is tested with a different size ranging in 10, 25, 75, and 300 nodes to emulate particular sensor network environments such as a room of a home to a sizable commercial office space. Stated before, topologies for the star and grid were predictable with the grid topology only differing with a varied number of columns. The mesh network was created with pseudo-random edges between nodes each time a topology

Figure 4: Effective Throughput

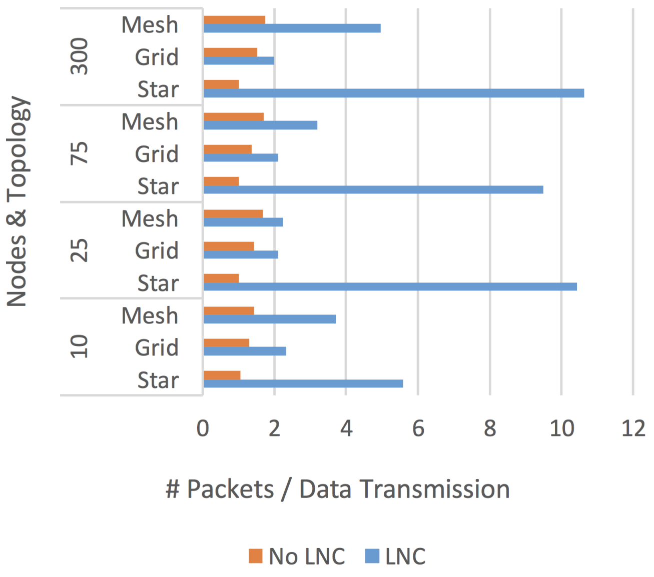
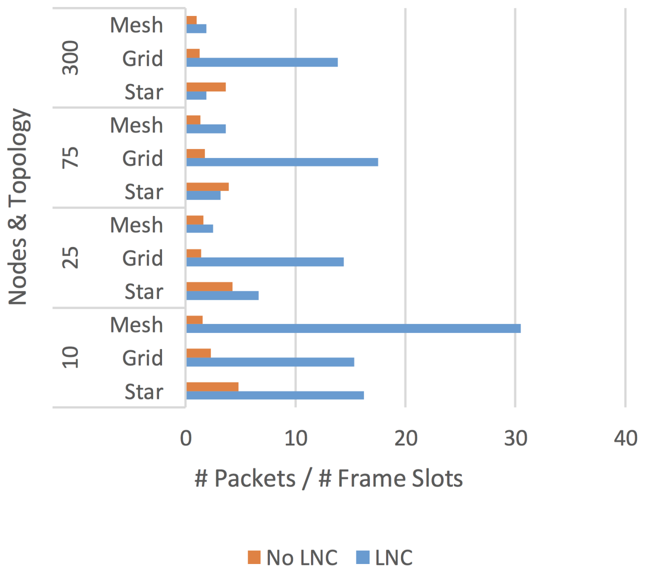


Figure 5: Throughput



was created.

For some of the packet parameters, our test will be based on the 802.15.4 standard packet payload size of 127 bytes. While the size itself will not be changed, some overhead will be added to more efficiently push packets through the network. However, if the overhead required to encode these packets takes too many bytes for a useful message to be sent, this would call for the size of the IEEE 802.15.4 packet to be increased in a network that utilizes linear network coding, which in turn need to be evaluated against a sensor node that has limited resources. Also to be considered, the packet encoding method will range from sending an entire packet to encoding many packets together and decoding them at the node. This will correlate closely with the above mentioned minimum packet size because if 127 bytes is not enough, packet encoding can be an effective and efficient solution to this problem. In testing the limits of packet destination, the range of number of destinations must be wide because of the variety in the predictable nature of the star topology that does not require complex networking for multicast packets to the unpredictable nature of the mesh network where an “efficient” multicast route requires more complex and intelligent utilization of network coding. Within the simulator, we start with a unicast packet to a maximum of a total flood, completely chosen at random by a packet generating script. To do complex routing of larger scale, we must test if linear network coding will have a positive impact as the number of nodes increases.

Each simulation is run twice, one running with and one running without linear network coding. The given results are latency (in ticks), total number of ticks to run a simulation (in ticks), ticks where there was network activity (in ticks), CTS’s emitted and received, RTS’s emitted and received, number of destinations, and number of ticks in which a packet reached a destination. With all of these results, we analyze the data to draw our conclusions.

5. RESULTS

Our first parameter, ”effective throughput”, was calculated as the ratio of packets to the number of frame slots during which data was sent. We designated this measurement ”effective throughput” because it is only considering the network layer. That is, it does not address the effects of implementing LNC with respect to the link layer. As shown in figure 4, using linear network coding (denoted ”LNC”) increased the effective throughput of every network type at every scale compared to not using linear network coding (”No LNC”). This increase in effective throughput allows us to make a definitive statement about the higher quality of the linearly encoded network.

Comparing the improvement linear network coding has on effective throughput with the classic definition of throughput, as is shown in 5, we observe similar results. Here, throughput improved again on all network topologies consisting of a maximum of 25 nodes.

However, throughput decreased on star topologies with at least 75 nodes. This is to be expected, as these high contention networks have less opportunity to leverage the performance advantages of LNC. Additionally, these results do not detract from the notion of LNC being effective because running a 75+ node network in a star topology (keeping in mind that in such a case all 75 nodes are within 20 meters of each other) is both unlikely and ill-advised.

Next we measured the latency of the different topologies. As seen in 6, linear network coding increased latency on all but the largest star topologies.

Effective load was defined as ratio of frame slots on which data was sent to the total number of frame slots. Meanwhile, the ratio of frame slots on which any data was sent to the total number of frame slots was designated ”load”. In 7 and 8, linear network coding greatly reduced effective load and load, respectively.

Figure 6: Latency

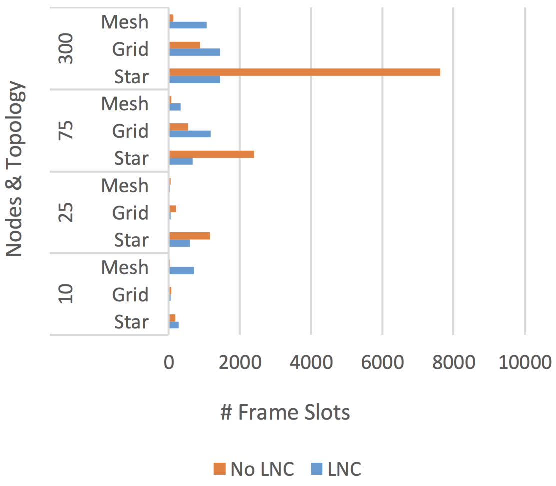


Figure 7: Effective Load

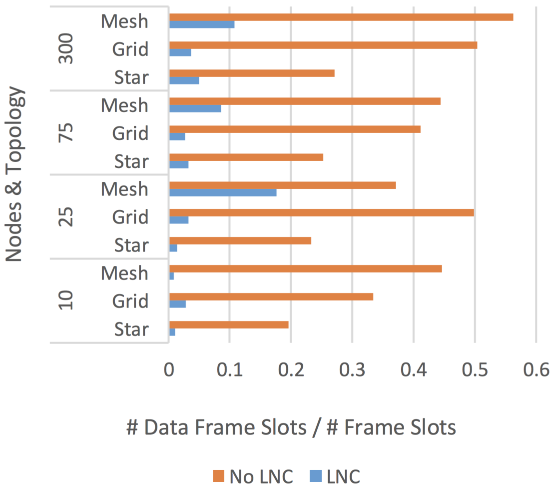
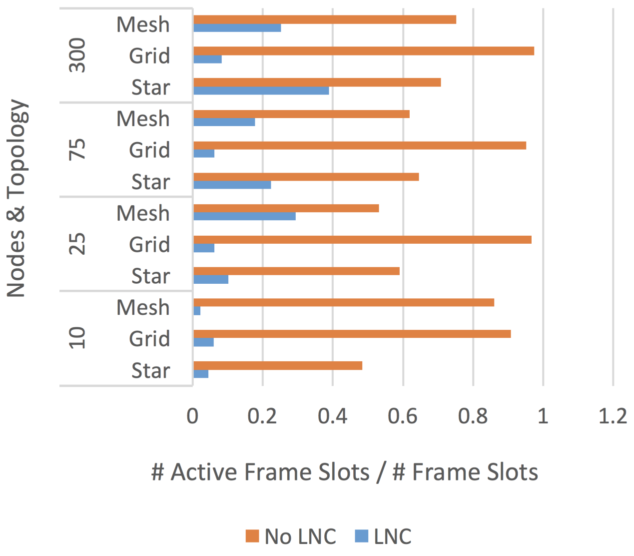


Figure 8: Load



6. CONCLUSIONS

Our network coding algorithm was capable of running within the constraints of the 802.15.4 protocol on a variety of network arrangements. For each network topology simulated, we proved the efficacy of implementing linear network coding as a means to improve throughput while simultaneously decreasing the need for routing. Our approach eliminates much of the computational complexities of LNC, and reduces encode/decode complexity to levels that can be achieved by the processors bound in the embedded devices found in wireless sensor networks. Our solution fits in with existing protocols and equipment, and can existing implementation can be easily extended to include our optimization. Our methodology is a viable approach to improve performance and power consumption, thereby lowering the costs associated with deploying, maintaining, and operating wireless mesh networks.

7. REFERENCES

[1] R. Ahlswede, Ning Cai, S. Y. R. Li, and R. W. Yeung. Network information flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, Jul 2000.

[2] N. Barroca, Luis M. Borges, F. J. Velez, and P. Chatzimisios. Ieee 802.15.4 mac layer performance enhancement by employing rts/cts combined with packet concatenation. In *IEEE International Conference on Communications - ICC*, volume -, pages —, June 2014.

[3] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft. Xors in the air: Practical wireless network coding. *IEEE/ACM Transactions on Networking*, 16(3):497–510, June 2008.

[4] J. Krigslund, J. Hansen, M. Hundeboll, D. E. Lucani, and F. H. P. Fitzek. Core: Cope with more in wireless meshed networks. In *Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th*, pages 1–6, June 2013.

[5] S. Sengupta, S. Rayanchu, and S. Banerjee. An analysis of wireless network coding for unicast sessions: The case for coding-aware routing. In *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, pages 1028–1036, May 2007.