Rateless Coding for Multi-Hop Broadcast Transmission in Wireless IoT Networks

Guo, Jianlin; Koike-Akino, Toshiaki; Wang, Pu; Parsons, Kieran; Orlik, Philip V.; Shanmuga Sundaram, Jothi Prasanna; Sumi, Takenori; Nagai, Yukimasa

TR2023-073 June 27, 2023

Abstract

The software distribution in advanced IoT net- works is inevitable. However, distributing software in multi-hop wireless networks consumes enormous communication band- width and can also suffer from reliability challenge. This paper proposes an innovative dynamic relay point (DRP) protocol to reduce the number of software packet transmissions. It also introduces a network condition based rateless coding scheme to improve the packet transmission reliability. The NS3 simulator is employed for performance evaluation. The proposed DRP protocol outperforms multi-point relay (MPR) baseline by reducing the software packet transmissions and improving the effective throughput.

IEEE International Symposium on Information Theory (ISIT) 2023

Rateless Coding for Multi-Hop Broadcast Transmission in Wireless IoT Networks

Jianlin Guo*, Toshiaki Koike-Akino*, Pu Wang*, Kieran Parsons*, Philip Orlik*,
Jothi Prasanna Shanmuga Sundaram*‡, Takenori Sumi†, and Yukimasa Nagai†
*Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139, USA
†Information Technology R&D Center, Mitsubishi Electric Corporation, Kamakura, Kanagawa 2478501, Japan
‡Electrical Engineering and Computer Science, University of California, Merced, CA 95343, USA

Abstract—**The software distribution in advanced IoT networks is inevitable. However, distributing software in multi-hop wireless networks consumes enormous communication bandwidth and can also suffer from reliability challenge. This paper proposes an innovative dynamic relay point (DRP) protocol to reduce the number of software packet transmissions. It also introduces a network condition based rateless coding scheme to improve the packet transmission reliability. The NS3 simulator is employed for performance evaluation. The proposed DRP protocol outperforms multi-point relay (MPR) baseline by reducing the software packet transmissions and improving the effective throughput.**

Index Terms—**Rateless coding, dynamic relay decision, multi-hop broadcast transmission, software distribution, lossy wireless networks.**

I. INTRODUCTION

As more and more intelligent devices connect to the Internet, Internet of Things (IoT) applications such as smart metering and environmental monitoring have been rapidly increasing. IoT devices are resource constrained and initially installed with the minimal software (and/or firmware), which is updated later as needed. Therefore, the software distribution in IoT networks becomes inevitable. In addition, the software update must be distributed to entire network since it is responsible for network operation.

The flooding plays an important role in multi-hop wireless networks, e.g., in route discovery. However, flooding leads to a large number of redundant packet transmissions that consumes enormous communication bandwidth and energy, and can cause collisions and thus packet loss. The multi-point relay (MPR) protocol [1] has been proposed to reduce the number of transmissions. Alternative methods such as the HopCaster [2] and the multi-hop broadcast protocol [3] have been also proposed. However, these static relay node selection mechanisms do not take dynamic wireless network condition into account.

Furthermore, packet transmission in lossy wireless networks such as smart meter network is not reliable. To tackle the reliability issue, the coding techniques have been applied, especially lightweight Luby transform (LT) codes [4] based coding schemes, whose data encoding and decoding processes do not significantly increase energy consumption. Authors in [5], [6] and [7] applies LT codes to MPR based multi-hop broadcast transmission for reliability improvement. Literature [8] proposes a fountain broadcast protocol to disseminate software in multi-hop wireless networks. However, none of these works has considered the dynamic wireless network properties such as link quality— a critical metric of the wireless networks— and the node condition.

Another issue to be addressed in broadcast transmission is the acknowledgement. In wireless communication protocols, broadcast transmission is not acknowledged. Therefore, there is no guarantee that a broadcast transmission is successfully delivered. To ensure the successful software distribution, the re-transmission may be needed. Without acknowledgement, a transmitter does not know whether the re-transmission is needed. Most importantly, the rateless coding also needs acknowledgement so that the encoder knows when to stop transmission. Yet the existing works do not address the acknowledgement issue.

In this paper, we address aforementioned challenges for software distribution in multi-hop wireless IoT networks. The main contributions of this work are summarized as follows:

- We propose an innovative dynamic relay point (DRP) protocol for broadcast transmission in multi-hop IoT networks to reduce the number of software packet transmissions and energy consumption.
- We introduce a rateless coding scheme to encode software packets by considering dynamic network condition for transmission reliability improvement without significant energy consumption increase.
- An acknowledgement mechanism is provided for broadcast transmission in multi-hop wireless networks.
- We also present a hybrid packet transmission method to balance the redundancy and the reliability.

II. RELATED WORKS

The MPR protocol aims to select a subset of the relay nodes to cover entire network. However, finding the smallest relay set is NP hard [1]. Accordingly, a heuristic relay node selection method has been provided, in which each node independently selects its 1-hop neighbors to cover its 2-hop neighbors. Alternative relay protocols have been also proposed. Work [2] presents a HopCaster method to build a multicast tree. However, the tree structure is not based on communication links but on logical routing links. Thus, a node can receive packets not only from 1-hop neighbors but also from 2-hop neighbors, which degrades the forwarding efficiency. A multi-hop broadcast protocol in [3] selects 2-hop downstream forwarders with 1-hop neighbors being forwarders. Yet the selection of 2-hop forwarders uses 3-hop

---

Jothi Prasanna Shanmuga Sundaram was working at MERL as an intern.

neighbor information, which makes the protocol not efficient but more complicated.

There are different coding techniques including network coding and rateless erasure coding. Although network coding such as random linear network coding (RLNC) [9] is more flexible, it has high complexity and thus, may not fit the resource constrained IoT devices. Accordingly, the lightweight rateless coding becomes ideal candidate for IoT devices.

LT codes [4] is a lightweight coding scheme due to its low complexity of XOR encoding and decoding. It is the base of rateless erasure coding and has been adopted in popular Raptor codes [10]. To generate an encoded packet, LT randomly chooses a degree d, i.e., number of source packets, from a degree distribution function. LT then chooses uniformly at random d distinct source packets as input packets. The value of the encoded packet is the XOR of the d input packets. The degree distribution function is the key in LT codes. The Ideal Soliton distribution and the Robust Soliton distribution are proposed in [4]. The alternatives such as the Switched distribution [11] and the optimized Robust Soliton distribution [12] have been also proposed. There are different decoding methods for LT coding. The Belief propagation and the Gaussian elimination are widely used LT decoding methods. The alternative decoding methods include on the fly Gaussian elimination [13]. The fountain broadcast protocol in [8] disseminates software in multi-hop wireless networks, in which a node blows up the m source packets into n encoded packets using LT encoding and the encoded packets are transmitted into the network. Each recipient re-transmits a new packet with a probability p. When a recipient node receives k encoded packets ($k \geq m$) as required for decoding, it reconstructs the source packets and passes them to the application. Authors in [5], [6] and [7] applies LT coding to MPR based multi-hop broadcast transmission in wireless networks. These works use heuristic relay node selection to reduce number of transmissions, in which the MPR relay nodes do not simply forward packets they overhear but also send out information that is encoded over the contents of several packets they received.

Works [2] and [3] apply network coding to improve broadcast transmission reliability, in which [2] applies physical layer modulation and channel coding schemes and [3] does not specify the coding scheme used. Literature [14] proposes a network coding assisted reliable multi-source multicasting protocol for multi-hop wireless mesh networks. It uses a multicast least cost anypath routing algorithm to select relay nodes and applies a binary network coding scheme for encoding. Authors in [15] introduces a probabilistic cooperative coded forwarding scheme for multi-hop data transmission in mobile edge industrial internet of things (IIoT) to improve transmission reliability and reduce packet delay. The systematic sparse network coding mechanism is used to encode packets.

III. SYSTEM MODEL

This paper studies software distribution in multi-hop wireless IoT networks, where network manager distributes the software program to all data nodes in the network via hop-by-hop relay. We consider a network consisting of a network manager and a set of data nodes forming a multi-hop topology.

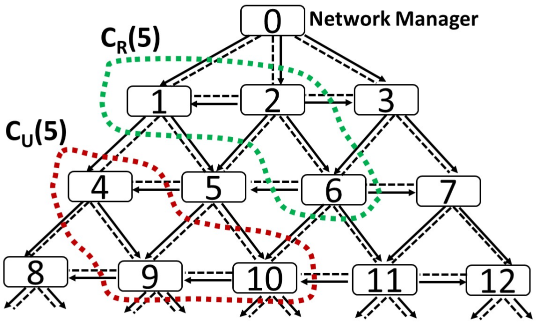

Fig. 1. Multi-Hop Broadcast Transmission Model in Wireless IoT Networks

Fig.1 illustrates our system model, where node 0 is the network manager, the solid lines represent the software distribution and the dash lines represent physical links. Unlike the logical link used in [2] and [3], where a node can communicates with the nodes that are multiple hops away, using physical communication links, the neighbors of a node are only nodes from which the node can receive packets.

There are different ways for data nodes to relay software packets, e.g., 1) relay packets when software program is completely received and 2) relay a software packet once a packet is received. This paper uses approach 1) since it gives the intermediate data nodes opportunity to re-encode software packets based on their conditions such as link quality.

IV. DYNAMIC RELAY POINT PROTOCOL FOR MULTI-HOP BROADCAST TRANSMISSION

We assume that a software program needs to be carried into K network packets and must be distributed to all data nodes.

A. *Dynamic Neighbor Classification*

This paper introduces a dynamic neighbor classification method, in which a node n divides its neighbors into Class-R set denoted as $C_R(n)$ and Class-U set denoted as $C_U(n)$, where $C_R(n)$ consists of all neighbors that have received software and $C_U(n)$ includes all neighbors that have not received software, i.e., the neighbors that have not successfully decoded the software. Fig.1 illustrates the Class-R set and Class-U set of node 5.

Initially, each node adds all neighbors except node 0 into Class-U set, e.g., node 1 adds nodes 2 and 6 into Class-U set but puts node 0 into Class-R set. On the other hand, node 2 adds all neighbors 1, 2, 4, 6, 9 and 10 into Class-U set. As the software distribution proceeds, each node learns neighbor's states by monitoring software packet transmissions from its neighbors. If a node overhears a software packet transmission or an acknowledgement from a neighbor, the node moves that neighbor from Class-U set into Class-R set.

It can be seen that Class-R set and Class-U set are two dynamic sets, whose members dynamically change as the software distribution progresses.

B. Passive and Active Acknowledgement

Applying rateless coding technique in broadcast transmission faces a particular challenge. On the one hand, wireless communication protocols such as IEEE 802 standards don't acknowledge the broadcast transmission. On the other hand, the rateless coding requires acknowledgement so that the encoder can stop transmission. The conventional rateless coding schemes are designed for end-to-end topology, in which the encoder continues packet transmission until an acknowledgement is received from the decoder. This is not the case in multi-hop broadcast relay, where a node broadcasts packets to all neighbors and also receives packets from all neighbors. Therefore, even if some neighbors acknowledge, other neighbors may still need packets. As a result, the conventional transmission stopping mechanism does not work.

To address this dilemma, we propose a hybrid acknowledgement mechanism, in which the transmission of a software packet serves as a passive acknowledgment and the transmission of an ACK packet serves as an active acknowledgement. When a node overhears a software packet or an ACK packet transmission from a neighbor, it considers that neighbor has successfully received the software and therefore, moves that neighbor from Class-U set to Class-R set. It should be noticed that a leaf node may have a non-empty Class-U set and therefore, need to relay software.

C. Dynamic Relay Point Protocol

Reducing the number of software packet transmissions and the software distribution latency is critical in multi-hop IoT networks. The routing protocol based upstream and downstream relay node selection such as the selection method in [3] is ambiguous in some cases, e.g., for node 2 in Fig.1, node 0 is a upstream node and node 6 is a downstream node, but it is difficulty to classify nodes 1 and 3.

This paper proposes an innovative dynamic relay point (DRP) protocol that makes relay decision based on dynamic network conditions. More specifically, when a data node successfully receives the software program, it relays software only if its Class-U set is not empty. Otherwise, it broadcasts an ACK packet.

The software relay is scheduled based on a proposed scheduling method, in which network manager 0 schedules software packets without delay. However, a data node n schedules software packets with the delay calculated as follows

$$D(n) = \frac{TimeUnit}{N_U(n) * \sum_{i=1}^{N_U(n)} q_t(i)}, \tag{1}$$

where $TimeUnit$ can be us, ms, etc., $N_U(n)$ is the number of Class-U neighbors and $q_t(i)$ is the link quality from node n to Class-U neighbor i at time t. Eq.(1) indicates that the scheduling delay is inversely proportional to the number of Class-U neighbors and link qualities to Class-U neighbors. If a data node has larger Class-U set with good link qualities, this node schedules software packets early so that the packets can be reliably delivered to more Class-U nodes, which will in turn relay software packets early to speed up the distribution. Otherwise, if a data node has smaller Class-U set or poor link qualities, this node schedules software packets later because fewer Class-U nodes can receive data packets from it or the packets may get lost due to the poor link quality. With the longer delay, its Class-U neighbors have potentials to receive software packets from alternative sources.

Using the proposed DRP protocol, a node has two classes of link qualities corresponding to two sets of neighbors, i.e., link qualities from Class-R neighbors to that node and link qualities from that node to Class-U neighbors. These two classes of link qualities are independent. It indicates that even a node has poor link quality to its Class-U neighbors, its Class-R neighbors may still have good link quality to distribute software to it.

The neighbor discovery can be done in different ways, e.g., using a specific neighbor discovery method or during a route discovery process. This paper assumes neighbor information is available. Similarly, the link quality can be expressed in different ways, e.g., ETX and RSSI, and measured differently, e.g., monitoring acknowledgment packet reception and measuring received signal strength. This paper assumes link quality information is also available.

V. NETWORK CONDITION BASED SOFTWARE PACKET ENCODING

To address issues in applying conventional end-to-end coding schemes to multi-hop broadcast transmission, this paper presents a network condition based software packet encoding for software distribution in multi-hop wireless IoT networks.

Upon successfully decoding K software packets, a data node relays software only if it has Class-U neighbor. It can re-encode these K packets based on the network conditions it has learned. It first generates $N(n)$ ($\geq K$) encoded packets, which can be enough to decode K source packets, and the additional packets are generated if needed. The $N(n)$ is calculated according to

$$N(n) = K * (1 + RD(n)), \tag{2}$$

where $RD(n)$ is the redundancy level computed as

$$RD(n) = \frac{C * N_U(n)}{1 + \min_{\forall i \in C_U(n)} q_t(i)}, \tag{3}$$

where C is a coefficient to take account of the link quality representation. LT encoding is then applied to generate $N(n)$ encoded packets, which are packed into network packets as payload together with header including degree d, coding neighbor vector, source packet number K, CRC checksum, etc. The network packets are then broadcasted into the network. Meanwhile, node n monitors active and/or passive acknowledgements from its Class-U neighbors. If $R\%$ of Class-U neighbors acknowledge the software reception, node n stops software packet transmission. If all $N(n)$ packets have been transmitted and the $R\%$ has not been reached, the node n will generate and broadcast additional packets.

The $R\%$ can be set based on number of Class-U neighbors and link qualities to Class-U neighbors. The $R\%$ stopping mechanism is proposed to reduce redundant transmissions by considering multi-source reception in broadcast transmission. It also takes network condition into account, e.g., if a node has poor link qualities to some Class-U neighbors, it tends to add more redundancy to increase the possibility of successful decoding, yet the added redundancy may not help. Stopping transmission avoids the scenario where the link quality is extremely poor and thus packet arriving rate is very low. In this case, keeping transmission consumes communication bandwidth without much gain. Stopping transmission gives other nodes opportunities to transmit, those nodes may have good link qualities to the poor link quality Class-U neighbors.

VI. SOFTWARE PACKET TRANSMISSION AND DECODING AS WELL AS REQUEST AND RESPONSE

This section introduces the proposed software packet transmission and decoding as well as request and response.

A. Hybrid Software Packet Transmission

Although coding techniques can improve transmission reliability, it adds redundancy as well. It has been shown that when the link quality is good enough, the simple transmission is more efficient than coding [16]. Therefore, if a node has good link qualities to all Class-U neighbors, it does not need to encode software packet. Accordingly, this paper proposes a two level hybrid transmission mechanism. The first level is about hybrid between software packet and ACK packet, where a node does not transmit software packet if it has no Class-U neighbor, instead it broadcasts an ACK packet. The second level is about hybrid between simple transmission and coding, in which a good link quality threshold is defined. If the link qualities to all Class-U neighbors are above the threshold, the node transmits software packets without encoding. Otherwise, the node transmits the encoded software packets.

B. Software Packet Decoding

A data node can apply any decoding method to decode the software packets. The decoding is attempted on the first K software packets received and each new software packet received later until K source packets are successfully decoded. Due to the hybrid transmission mechanism and multi-source reception, a data node may receive the encoded software packets from some Class-R neighbors and receive the non-encoded software packets from other Class-R neighbors. The non-encoded packets are treated as degree 1 encoded packets.

C. Timer Based Request and Response

To ensure software is distributed to entire network, this paper provides a timer based request and response mechanism. Upon receiving the first K encoded software packets, a data node starts a timer for re-transmission request if it does not successfully decode K software packets. When the timer is up, if the node still has not successfully decoded K source packets, it selects a Class-R neighbor and sends a re-transmission request. Upon receiving the request, the Class-R neighbor broadcasts the requested packets. Even the request is unicasted, the response is boradcasted to benefit more nodes. There are different ways to select a Class-R neighbor, e.g., selecting a Class-R neighbor with the best link quality or selecting a Class-R neighbor based on a routing protocol such as the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL).

VII. PERFORMANCE EVALUATION

In this section, we present our simulation results and performance analysis of the proposed techniques.

A. Simulation Settings

We use NS3 simulator with IEEE 802.15.4 communication protocol in 920 MHz band to evaluate the proposed technologies. The PHY data rate is set to 100 kbps. With these settings, the communication range in NS3 simulator is up to 230 meters.

In the simulation, we distribute a software of 10000 bytes. The proposed DRP protocol is evaluated in five aspects: (i) total number of software packet transmissions, (ii) total number of request packet transmissions, (iii) effective throughput per software packet transmission, (iv) overhead per software packet transmission and (v) total overhead to total effective throughput ratio. Both grid node deployment and random node deployment are simulated.

The MPR protocol is used as the baseline for performance comparison, in which each node uses heuristic method to select relay nodes. The software data is divided into 100 packets with payload size of 100 bytes. A node sends retransmission request if any packet is not received.

The DRP protocol is simulated using the proposed network condition based coding scheme. The software data is divided into 20 sessions of size 500 bytes. Each session is divided into 5 packets with payload size of 100 bytes. The distribution starts from session 1, next session starts when the current session completes. The Robust Soliton distribution and the Gaussian Elimination are employed for encoding and decoding. For each session, the Gaussian Elimination is called when the number of the encoded packets received is greater than or equal to 5. The $R\%$ is set to 80%. The CRC16 checksum of the encoded packet payload is carried in the packet header. In addition, NS3 provides an integer link quality from 0 to 255. We set hybrid transmission threshold to 200.

The RPL protocol is implemented for neighbor discovery and the re-transmission request, in which a data node sends re-transmission request to its default parent. The retransmission request timer is randomly set from 0 to 100ms.

B. Grid Node Deployment

The nodes are deployed in the first quadrant grid with the node 0 located at location $(0,0)$. The grid distances simulated are $25, 50, 100$ and 200 meters. The number of data nodes simulated is 1, 10, 25, 50, 100 and 150.

In the figures below, black lines, red lines, green lines and blue lines show simulation results for $25, 50, 100$ and 200 meter grids, respectively. Solid lines represent the MPR results and the dash lines represent the DRP results.

1) Total Number of software Packet Transmissions: The number of software packet transmissions is an important metric that have been addressed by existing works. The proposed DRP protocol outperforms the MPR baseline as shown in Fig.2. The DRP significantly reduces the number of software packet transmissions. For $25, 50, 100$ and 200 meter deployments, DRP can reduce software packet transmissions by up to 94%, 62%, 34% and 65%, respectively.

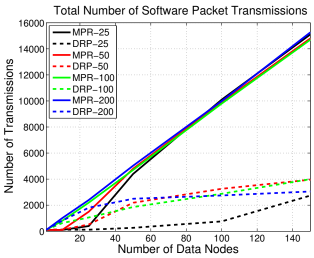

Fig. 2. Total Number of Data Packet Transmissions

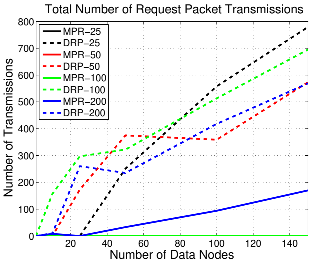

Fig. 3. Total Number of Request Packet Transmissions

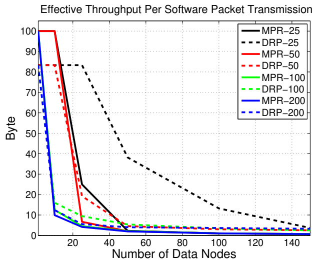

Fig. 4. Effective Throughput per Data Packet Transmission

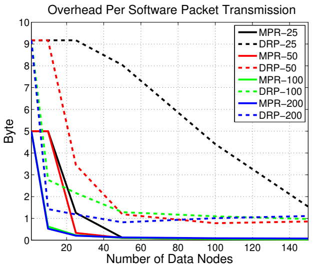

Fig. 5. Overhead per Data Packet Transmission

2) Number of Request Packet Transmissions: Although the DRP transmits less software packets, Fig.3 shows that the DRP transmits more request packets. Unlike the MPR that simply transmits source packets, the DRP transmits randomly encoded packets. When a data node receives K encoded packets, it is not guaranteed that the node can successfully decode K source packets. Therefore, the retransmission request is more likely needed. In addition, the MPR transmits much more software packets than the DRP does, which indicates that the re-transmission is less likely needed. In the denser $25, 50$ and 100 meter deployments, the MPR does not transmit request packet since data nodes have more opportunities to receive high link quality packets. However, in the sparser 200 meter deployment, the MPR transmits request packets because as grid distance increases, the communication link quality decreases and therefore, the re-transmission is needed.

3) Effective Throughput per Software Packet Transmission: The effective throughput, i.e., network packet payload, is another important metric that reflects the efficiency of the networking protocols. As shown in Fig.4, the MPR performs better than the DRP only when number of data nodes is less than or equal to 10 in 25 and 50 meter deployments, which are 1-hop networks without requiring relay. In such cases, the MPR only transmits source packets and the DRP, however, needs to transmit redundancy due to the coding. The DRP performs better in all multi-hop scenarios. For $25, 50, 100$ and 200 meter deployments, the DRP can improve effective throughput by up to 1579%, 198%, 241% and 401%, respectively. The DRP with 25 meter deployment performs best due to the least data packet transmissions. For both MPR and DRP, the effective throughput decreases as the number of data nodes increases due to the fact that same piece of software data needs to be relayed by more data nodes.

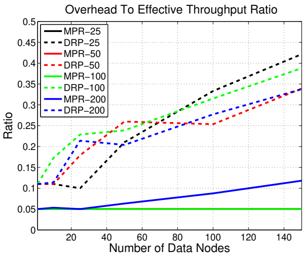

Fig. 6. Overhead to Throughput Ratio

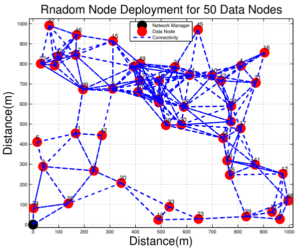

Fig. 7. Random Data Node Deployment

4) Overhead per Software Packet Transmission: The overhead includes software packet header and the content of request packet. The MPR outperforms the DRP in overhead category as shown in Fig.5. The DRP with 25 meter deployment has the highest overhead due to the most request transmissions. For both MPR and DRP, the overhead decreases as the number of nodes increases due to the increase of software packet transmissions. As a result, the overhead per software packet transmission decreases.

5) Overhead to Effective Throughput Ratio: Due to the smaller software packet header and less request packet transmission, the MPR outperforms the DRP in terms of the overhead to effective throughput ratio as shown in Fig.6. For $20, 50$ and 100 meter deployments, the MPR maintains 5% ratio due to no request packet transmission. The overall ratio ranges from 5% to 12%. For the DRP, the overall ratio ranges from 11% to 42%.

C. Random Node Deployment

In this case, 50 data nodes are randomly deployed in a 1000 meter by 1000 meter square area with node 0 at location $(0, 0)$ as shown in Fig.7. The MPR transmits 5126 software packets and 28 request packets. The effective throughput and overhead per software packet transmission is 1.95 bytes and 0.119 byte, respectively. The overhead to effective throughput ratio is 0.0612. On the other hand, the DRR transmits 1382 software packets and 273 request packets. The effective throughput and overhead per software packet transmission is 7.24 bytes and 1.586 bytes, respectively. The overhead to effective throughput ratio is 0.219. The DRP reduces software packet transmissions by 73% and improves effective throughput by 271%.

VIII. CONCLUSION

Distributing software in multi-hop lossy wireless networks faces challenges due to high communication bandwidth requirement and low transmission reliability. This paper proposes a DRP protocol for multi-hop broadcast transmission by making dynamic relay decision. It also introduces a network condition based rateless coding scheme for transmisson reliability improvement. The NS3 simulator is used for performance evaluation. The DRP protocol with proposed coding scheme outperforms the MPR baseline in two critical metrics by reducing software packet transmissions up to 94% and improving effective throughput up to 1579%.

REFERENCES

[1] A. Qayyum, L. Viennot, and A. Laouiti, “Multipoint relaying: An efficient technique for flooding in mobile wireless networks,” in *The 35th Annual Hawaii International Conference on System Sciences*, 2001.

[2] R. Hallousha, H. Liub, L. Dongc, M. Wud, and H. Radhae, “Hop-by-hop Content Distribution with Network Coding in Multihop Wireless Networks,” *Digital Communications and Networks*, vol. 3, no. 1, pp. 47–54, 2017.

[3] F. Yan, X. Zhang, and H. Zhang, “Efficient Multihop Broadcasting With Network Coding in Duty-Cycled Wireless Sensor Networks (NET),” *IEEE Sensors Letter*, vol. 1, no. 5, 2017.

[4] M. Luby, “LT Codes,” in *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2002.

[5] N. Kadi and K. A. Agha, “Optimized MPR-based flooding in wireless ad hoc network using network coding,” in *1st IFIP Wireless Days*, 2008.

[6] K. A. Agha, N. Kadi, and I. Stojmenovic, “Fountain Codes with XOR of Encoded Packets for Broadcasting and source independent backbone in Multi-hop Networks using Network Coding,” in *IEEE VTC Spring 2009*, 2009.

[7] N. Kadi and K. A. Agha, “MPR-based flooding with distributed fountain network coding,” in *2010 The 9th IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2010.

[8] R. Kumar, A. Paul, and U. Ramachandran, “Fountain broadcast for wireless networks,” in *IEEE International Workshop on Network Sensing Systems*, 2005.

[9] T. Ho, M. Medard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A Random Linear Network Coding Approach to Multicast,” *IEEE Transactions on Information Theory*, vol. 12, no. 10, 2006.

[10] A. Shokrollahi, “Raptor Codes,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[11] N. Kadi and K. A. Agha, “New degree distribution to improve LT-code in network coding for broadcasting in ad-hoc wireless networks,” in *21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2010.

[12] Q. Du, X. Song, Y. Liu, and L. Zhao, “Implementation of LT Code with a Novel Degree Distribution,” *Open Journal of Applied Sciences*, vol. 2, no. 4B, pp. 2551–2567, 2012.

[13] V. Bioglio, M. Grangetto, R. Gaeta, and M. Sereno, “On the Fly Gaussian Elimination for LT Codes,” *IEEE Communications Letters*, vol. 13, no. 12, 2009.

[14] P. Rathore, K. Dhaka, and S. K. Bose, “Network coding assisted reliable multi-source multicasting over a multi-hop wireless mesh network,” *Computer Communications*, vol. 172, pp. 130–141, 2021.

[15] W. Huang, X. Li, and Y. Liang, “Probabilistic cooperative coded forwarding for broadcast transmissions in industrial mobile edge communications,” *Journal on Wireless Communications and Networking*, 2022.

[16] H. Wen, C. Lin, F. Ren, Y. Yue, and X. Huang, “Retransmission or Redundancy: Transmission Reliability in Wireless Sensor Networks,” in *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, 2007.