A Link-Correlation-Aware Cross-Layer Protocol for IoT Devices

Fangming Chai*, Ting Zhu†, Kyoung-Don Kang*
*State University of New York at Binghamton
{fchai1, kang}@binghamton.edu
†University of Maryland, Baltimore County
zt@umbc.edu

***Abstract*—The Internet of Things (IoT) applications is envisioned to require higher throughput protocols because of the increasing data amount. To significantly enhance the network throughput between IoT devices, this paper proposes a new link-layer data forwarding technique that is aware of link correlation (LC) and supports receiver initiated acknowledgement (RI-ACK). We also propose a multicast communication protocol based on LC-aware forwarding and RI-ACKs to further enhance the throughput. In a simulation study, our protocol improves the throughput by 35% - 55% comparing to a state-of-the-art baseline.**

I. Introduction

The Internet of Things (IoT) mainly aims to connect various things, e.g., phones, tablets, sensor, actuators, cars and other mobile devices, through existing physical network infrastructures, including WiFi, Bluetooth, 802.15.4, Z-wave, and LTE-Advanced [1], [2]. Devices are connected together and cooperate to achieve some common goals, e.g., transportation management and mHealth (mobile health). Among the devices, some are low-rate low-power devices such as sensors and actuators, while others are normal-power high-rate devices like phones and cars. IoT communications can be classified into human-to-human (H2H), things-to-human (T2H), and things-to-things (T2T) communications [3].

In the future, it is envisioned that an increasing number of things will provide similar data accessibility to the accessibility provided by servers on the Internet[4]. Thus, providing stable, scalable, and high-throughput connections between things is essential in IoT. Further, support for not only one-hop but also multi-hop communications is required for IoT [1], [2].

Most IoT communication protocols have evolved from wireless sensor networks (WSN), machine to machine (M2M) or device to device (D2D) [3] communications. In these fields, effective communication protocols have been developed [5], [6], [7], [8], [9], [10]. However, the protocols from cellular networks, such as D2D or M2M [7], mainly support unicast data transmissions designed for a centralized topology. Many protocols in WSN and M2M are only designed to support unicast [8], [10]. In wireless networks, throughput is highly affected by the utilization of the physical spectrum. Although wireless communications have a broadcasting nature, a unicast transmission monopolizes and, thereby, underutilize the channel. As a result, throughput is decreased substantially.

Based on these observations, we have found opportunities to enhance the wireless network performance for IoT. First, we observe that most existing MAC layer protocols in IoT wireless networks are designed without considering the wireless link quality. In reality, however, different wireless links and networks may interfere with each other. Even though effective routing or similar solutions between devices greatly improve the efficiency, a cross-layer design that considers the link quality and correlation can provide better performance. It is important to design a scalable protocol that utilizes the broadcast nature of wireless networks, while considering wireless link qualities and correlations [11].

In this paper, we design a new IoT protocol to improve the throughput of an IoT network via effective link-layer data forwarding and multicast techniques. Our contribution includes:

1) We propose a new link-layer protocol that is aware of link correlation (LC) and leverages a receiver initiated acknowledgement (RI-ACK).

2) We design a new cross-layer protocol that supports multicast and multi-hop communications to enhance the packet transmission rate by leveraging the LC-aware forwarding and RI-ACK schemes.

3) We evaluate the performance of our protocol via a simulation study. Comparing to a state-of-the-art protocol that combines CSMA and OSPF, our protocol improves the throughput by 35%-55% and generally enhances goodput and fairness.

The rest of the paper is organized as followed. Section II and III give background for RI-ACK and LC-aware packet forwarding and describe our protocol design. The performance evaluation is undertaken in Section IV. Related work is dis cussed in Section V. Section VI concludes the paper and discusses future work.

II. Packet Forwarding Aware of Link Correlation

In this section, we introduce the LC-forwarding and RI-ACK mechanisms. l_{ij} and $P(i|j)$ indicate the link quality from node i to node j and the conditional packet reception probability of node i given that node j receives the same packet [11], respectively.

A. Receiver Initiated Acknowledgement

Figure 1 shows how RI-ACK works. A sender sends a packet to multiple receivers which form a transaction group. Receivers listen to the channel and receive the packet. Only the receiver to which the packet is destined but missed the packet will initiate an acknowledgement (ACK). All the other receivers in the transaction group, which successfully received the packet, are potential candidates to forward the packet. The receiver that missed the packet sends an ACK to the preferred candidates picked from the group to request the missing packet. It assigns priorities to the candidates such that, in the RI-ACK header, their addresses appear in priority order. The highest priority candidate that receives the packet wins the opportunity to forward the packet.

In this way, an RI-ACK allows a receiver to select its neighbor to forward the missing packet rather than requesting the original sender to re-transmit the packet. There are two benefits of RI-ACK. First, an ACK is only delivered on demand. Let l_{sr} denote the link quality from the sender s to receiver r. If $l_{sr} > 0.5$, the receiver is more likely not to transmit an ACK. The higher l_{sr} is, the less likely for the receiver to transmit an ACK. Thus a RI-ACK mechanism considerably reduces the number of ACK packets that consume the precious wireless bandwidth. It forwards packets considering the current network state. Also, when r misses a packet, it first checks nearby candidates, which may hold the missing packet, and tries to get the packet from the candidate with the best throughput before contacting s.

B. Link-Correlation Aware Forwarding

In this paper, we use expected transmission count (ETX) [12], [13] to estimate the link quality of different candidates. Also, we define expected transmission time (ETT) as followed:

$$ETT = \frac{L * ETX}{R} \tag{1}$$

where L is the packet length and R is the channel transmission rate.

For a reliable transmission including data and ACK without considering the link quality and correlation, the ETT from node s to r is the sum of the ETTs for the data and ACK:

$$ETT = \frac{L_{data} * ETX_{sr}}{R} + \frac{L_{ACK} * ETX_{rs}}{R} \tag{2}$$

With link quality awareness, the receiver can request a missed packet from any of the candidates. If it requests candidate node q to transmit the packet, the ETT is:

$$\begin{aligned} ETT = (1 - l_{sq}) * (\frac{L_{data} * ETX_{sr}}{R} + \frac{L_{ACK} * ETX_{rs}}{R}) \\ + l_{sq} * (\frac{L_{data} * ETX_{qr}}{R} + \frac{L_{ACK} * ETX_{sq}}{R}) \end{aligned} \tag{3}$$

By leveraging link correlation, equation 3 is re-written as:

$$\begin{aligned} ETT = P(q|\bar{r}) * (\frac{L_{data} * ETX_{sr}}{R} + \frac{L_{ACK} * ETX_{rs}}{R}) \\ + P(\bar{q}|\bar{r}) * (\frac{L_{data} * ETX_{qr}}{R} + \frac{L_{ACK} * ETX_{sq}}{R}) \end{aligned} \tag{4}$$

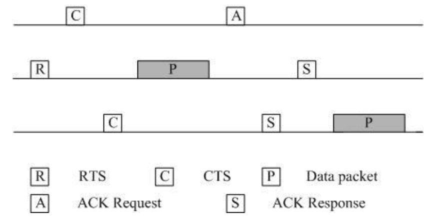

Fig. 1. Packet sequence

Here we use $P(i|\bar{j})$ to represent negative link correlation that is the probability for node i to get a packet in case node j does miss it. Assuming L_{data} is always larger than L_{ACK}, [11] showed that the link correlation can be bigger than the link quality.

When a node misses a packet and LC-aware forwarding happens, it will get the packet from the candidate with the minimum ETT:

$$ETT = \min_{i \in N} ETT_{ir} \tag{5}$$

where N is the set of the IDs of r's neighbors, which may have received the packet. This is because receiver r enumerates them in the RI-ACK in descending priority order as discussed before.

III. Protocol Design

In this section, we design a cross-layer IoT protocol − mainly across the MAC and network layers − by utilizing link correlations and RI-ACKs.

A. Overview

The protocol utilizes RTS and CTS as the MAC layer synchronization scheme. Packets to different receivers are aggregated as a single packet to the receivers. Transmitting the aggregated packet to the receivers is performed as one transaction. To transmit an aggregated packet to multiple nodes, an RTS message is formed to contain the destination MAC addresses. Any node that receives the RTS and has a matching MAC address should send a CTS frame to confirm the transmission as shown in Figure 1. The sequence to send CTS is the same as the MAC address sequence in the RTS frame as shown in lines 3 and 4 in Algorithm 1 that summarizes our protocol. Every node whose address is included in the RTS forms a transaction group TG as described in line $5-8$ in Algorithm 1.

The protocol utilizes frame aggregation to reduce the number of control frames. Aggregated frames whose format is shown in Figure 2 are linearly composed of simple frames to different receivers. There are many packet coding techniques [14], [21] to reduce coded packet length but they are highly constrained by either the network topology or specific use scenarios. Although aggregation isn't as efficient as network coding, it can be used without any restriction. However, packets to the same receiver won't be aggregated, because it only increases overall latency in the network.

In our protocol, a sender transmits an RTS frame to multiple receivers to send an aggregated packet that combines more

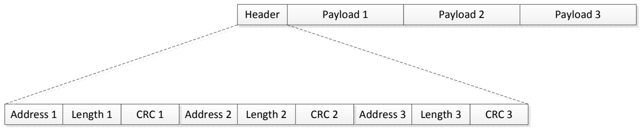

Fig. 2. Packet aggregation

than one frame. Destination nodes respond by transmitting CTS frames. The sequence of CTS frame transmissions is determined by the sequence of destination MAC addresses in the RTS frame. The sender transmits aggregated frames to the nodes that responded. After the sender transmits the aggregated frame, the transaction completes.

1) One Hop Network Aggregation: A side effect of aggregation is the increased duration of channel occupation. It is more significant if receivers aren't in their one hop range. To eliminate the side effect, we have to only aggregate packets to the perfect receivers which have identical neighbors. In practice, however, it is too ideal and restrictive to pick up perfect receivers only. Instead, we aggregate packets for similar neighbors. To measure the neighbor similarity of different nodes in a network, we introduce receiver similarity as below.

$$s_{ij} = \frac{number\ of\ common\ receivers}{number\ of\ total\ receivers} \tag{6}$$

Thus, if s_{ij} is closer to 1, nodes i and j have more common receivers.Packets to nodes i and j are aggregated if s_{ij} is equal to or higher than a specified threshold.

B. Receiver Initiated Acknowledgements in MAC

When an IoT device r receives a packet, it listens to the other nodes in the transaction group as described in lines $10-20$ in Algorithm 1. If r gets any request for a packet and r is the first node which acknowledges the request, r forwards the packet toward the receiver as described in lines $15-19$.

On the other hand, a receiver considers that it has missed a packet, if it overhears a frame with a larger ID or it receives an error checksum frame. Lines 22 to 32 in Algorithm 1 show the action taken by a receiver upon a packet miss. First, it adds all potential candidates, i.e., the CTS senders, to the request queue. It sorts the queue based on the ETTs and then sends the RI-ACKs to the nodes from the beginning of the request queue to the end. The last receiver in the request queue is the original sender s.

C. Lightweight Estimation of Link Quality and Correlation

All the nodes in the design maintain a table of link quality and link correlation information. A device estimates link quality by sending frames to and receiving them from its neighbors. To estimate link correlation, a node overhears the physical channel to get the information about its neighbors, while avoiding to explicitly exchange the information using extra control packets.

D. Design of IoT Routing

Our data link layer design can work effectively with any regular routing protocols and opportunistic routing protocols

Algorithm 1: Frame sequence

```
input : Sender s
input : Receiver r
input : Link quality Information
input : Link correlation Information
1  Transaction Group TG = Queue()
2  if rts = r.receive_rts_from(s) then
3      i = rts.index_of_dest_address_array(r.mac_address);
4      cts.send() in i^th slot ;
5      while n in rts.dst_address() do
6          if cts = r.receive_cts_from(n) then
7              TG.enqueue(n) ;
8          n++ ;
9  s sends an aggregated frame f ;
10 if r.receive(f) == True then
11     while n in TG do
12         if ack = r.receive_ack_from(n) and
           r.mac_address in ack.dst_address_array then
13             i = ack.index_of_dst_addr_array(r.mac_address) ;
14             win_to_forward = True ;
15             while k in ack.dst_mac_address_array do
16                 if r.receive_ack_response_from(k) then
17                     win_to_forward = False ;
18             if win_to_forward then
19                 r.send_simple_frame_to(n) ;
20         n++ ;
21 else
22     ACKQueue = Queue() ;
23     ACKQueue.enqueue(s) ;
24     while n = TG.dequeue() do
25         if ETT_n > ETT_s then
26             ACKQueue.enqueue(n) ;
27     sort(RequestQueue) to descending order by ETT ;
28     ack = new_packet() ;
29     while re==ACKQueue.dequeue() do
30         ack.add_dst_address(re.mac_address) ;
31         if re == s then
32             break ;
```

[13], [14], [17]. In our design, when a packet arrives at a non-destined node h, h either re-routes the packet to its IP destination d or forwards it to its link layer next-hop r if h receives RI-ACK from r as shown in lines 2 to 5 of Algorithm 2. A re-routing happens only when h has a better path to d than r does, and h notifies r in the RI-ACK response to suppress r to route the packet upon re-routing. Otherwise, h competes to forward the packet as described in section II-A. Our re-routing competition algorithm uses the priority order indicated in an

Algorithm 2: Opportunistic routing

```
input : Packet p
input : Original Receiver r
input : Packet Destination d
input : Packet holder h
1 if h receives RI − ACK from r then
2     if EXT_hd > EXT_rd then
3         h.compete_to_reroute()
4     else
5         h.compete_to_forward()
6 else
7     h.drop(p)
```

TABLE I
PARAMETERS IN THE SIMULATION

Name	Value
CRC	4B(Bytes)
payload	36B
Source MAC Address	6B
Destination MAC Address	6B
PHY Transmission rate	19.6Kbps[10]
Interframe Interval	0.03048us
Receiver similarity threshold (Eq. 6)	0.7
Maximum number of packets for aggregation	8

RI-ACK frame, similar to the forwarding competition method described in section II-A. If h doesn't get an RI-ACK request from r, h should drop the packet to avoid packet storm as specified in line 7 in Algorithm 2 .

IV. Performance Evaluation

The performance of the proposed protocol is evaluated via a simulation study. We have compared the performance to the CSMA MAC protocol and the OSPF (Open Shortest Path First) routing protocol [18] in both single-hop and multi-hop settings. In the 1-hop simulation, all nodes are in a one-hop network and, therefore, there are no hidden terminals. In the multi-hop simulation, we have 32 nodes each of which transmits packets to random receivers at a certain rate. Hidden terminal and exposed terminal are simulated in the multi-hop simulation.

Table I lists the parameters we used in the simulation. Unless specified otherwise, we use the parameters used in [10].

A. Throughput

In this experiment, we measured and compared the throughput and goodput [19], [20] of our protocol and CSMA. In 1-hop scenario, we varied the number of nodes in the network. Each node transmitted with its full capacity. Thus, the network was fully loaded regardless of the number of the nodes in the network. Figure 3 shows the throughput of our protocol in a 1-hop network. It shows that the throughput of our approach is better than CSMA by 35 - 55%. This improvement is mainly because our protocol greatly reduces control packets, i.e., RTS/CTS and ACK packets, between senders and receivers comparing to CSMA.

Figure 4 shows the goodput of our protocol in a 1-hop network. Usually, goodput is higher in a high throughput MAC protocol. Comparing to requesting packets from the original senders, getting packets from potential neighbors achieves better performance even in a 1-hop network. Even in a 1-hop network, our LC-aware packet forwarding converts some of the 1-hop transmissions to 2-hop transmissions to enhance the throughput.

Figure 5 shows the throughput performance of the proposed protocol in a multi-hop network. We have 32 nodes in the simulation. If every node generates one packet per second, it already exceeds the network transmission capacity. The result also shows that the multi-hop throughput performance of our approach is higher than that of CSMA MAC. In addition to reducing control packets, our LC-aware packet forwarding enhances the throughput in a multi-hop network. Using the receiver similarity threshold, our protocol picks candidates to form a transaction group among potentially receivers without aggravating hidden terminal problems. The OSPF routing protocol uses fixed paths to transmit packets; however, our underlying data forwarding method decides paths in real-time to select the near optimal ones considering link quality and correlation. After packets gets delivered using better paths, path selection is performed by OSPF. The result shows that our optimal path selection technique not only provides good performance but also works well with existing routing protocols, e.g., OSPF, as an essential supplement to routing in wireless networks. Therefore, in our cross-layer design, the MAC and routing layers effectively cooperate with each other. As shown in Figure 6, our approach is efficient in terms of goodput in multi-hop networks too.

B. Fairness

Many existing protocols support sender initiated fairness (SI-Fairness) such that senders have equal opportunities to transmit their packets. However, the number of packets transmitted to a common receiver from different senders can be largely different due to the diversity of link quality. To shed light on this issue, we introduce the receiver initiated fairness (RI-fairness) metric to measures the number of packets a node receives from its neighbors in a given time interval. General fairness is mathematically defined as equation 7.

$$f(n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n * \sum_{i=1}^{n} x_i^2} \quad (7)$$

For SI-fairness x_i is the number of packets node i sends out in the time interval and n is the total number of the nodes. For the RI-fairness, x_i is the number of packets a receiver successfully receives from each sender i and n is the total number of neighbors of the receiver.

In this experiment, we analyzed fairness of our protocol and CSMA. One of our design goals is to improve the RI-fairness without impairing SI-fairness. In the simulation, every node have equal weight to send packets. In CSMA, an ACK is

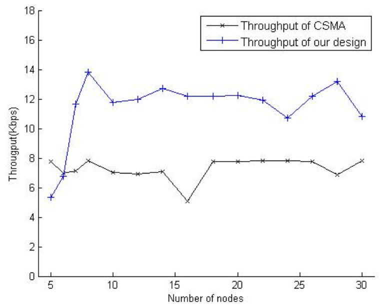

Fig. 3. Data Throughput in single hop network

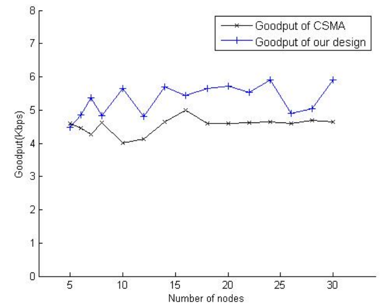

Fig. 4. Data goodput in single hop network

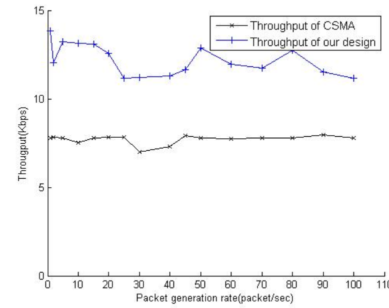

Fig. 5. Data Throughput in multi-hop network

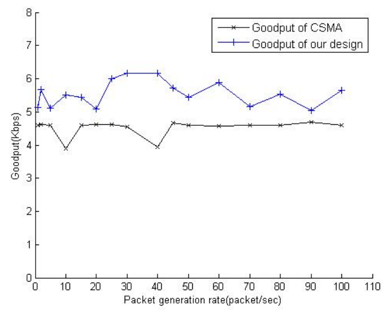

Fig. 6. Data goodput in multi-hop network

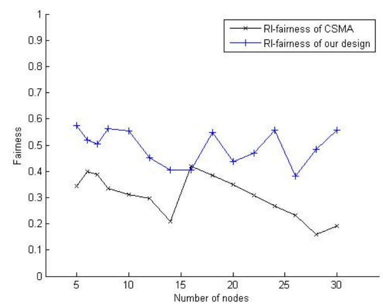

Fig. 7. RI-fairness of single hop network

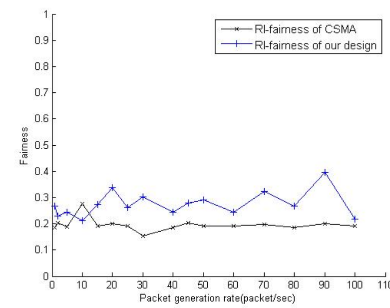

Fig. 8. RI-fairness of multi-hop network

delivered upon receiving a packet. However, in our approach, an RI-ACK is initiated only upon a packet miss as discussed before.

Figure 7 and Figure 8 show the RI-fairness in a 1-hop and a multi-hop networks, respectively. In a 1-hop network, the RI-fairness of CSMA ranges between $0.2 - 0.4$, while ours ranges between $0.4 - 0.6$. Thus, it enhances the RI-fairness by 20%-50%. In a multi-hop network, our approach outperforms CSMA on RI-fairness by 0.1. Figure 9 shows that our design has no adverse impact on SI-fairness in multi-hop networks. We also simulated single-hop SI-fairness in lab and the result is the exactly same as Figure 9 .

C. Sensitivity to Parameter Values

In this subsection, we analyzed the sensitivity of our protocol to parameter values. Our protocol has two parameters to set: 1) the maximum number of packets for aggregation, M_a, and 2) receiver similarity threshold, S_θ. In previous experiments, they were set to default values, 8 and 0.7 (Table I), respectively. To evaluate how the two variables impact the performance of our protocol, we evaluated the performance using different parameters.

Figure 10 shows the performance of the design for different M_a values. In the figure, the throughput gradually increases with M_a. As more packets are aggregated, fewer RTS/CTS frames are transmitted and more candidates are available for the packet forwarding from which a receiver can select a better path to get its missing packets. Besides, we observed M_a had no impact on the RI-fairness or SI-fairness. The simulation results are omitted due to space limitations.

Finally, we evaluated the performance of our protocol for different S_θ values. As described in Section III-D, the receiver similarity index has impact on the throughput.Figure 11 shows the throughput and goodput and confirms the aforementioned assertion. Throughput reaches maximum when $S_\theta = 0.7$ and then slightly drops. The reason is as S_θ goes up, it is more unlikely to pick receivers which can form a transaction group by equation 6. Similar to M_a, we observed S_θ had no impact on the fairness, either.

V. Related Works

Link correlation was first explored in [11], [22] in wireless sensor networks and then applied to common wireless networks [13]. The current research focus of link correlation is developing network-layer protocols e.g. flooding protocols [11] and opportunistic routing [13] etc. One of the challenges to utilize link correlation is that the correlation is weak in real wireless networks. As a result, most existing approaches [11], [13] featured opportunistic performance improvement. In contrast, we have built a general link correlation and receiver similarity model to improve wireless network performance across the data link and network layers. In addition, we extend opportunistic routing [13], [17], [23] to enhance path selections.

In wireless sensor networks (WSN), research has been done to improve the throughput [10], [24], [25], [26]. Z-MAC [10] reduced transmission collisions by dynamically switching between TDMA and CSMA. New approaches for low-power wireless bus [25], [26] investigated cross-layer protocols that utilized time-domain multicast to improve network performance. However, little work has been done to support space

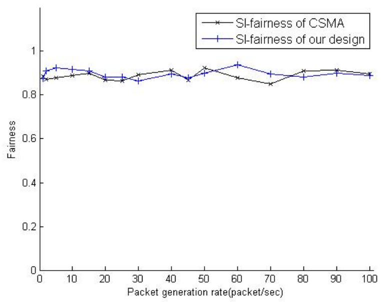

Fig. 9. SI-fairness of multi-hop network

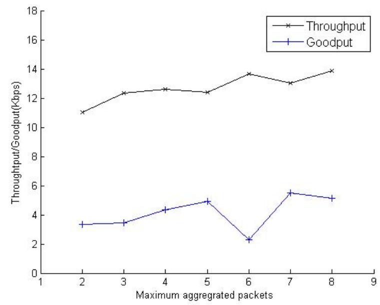

Fig. 10. Throughput and goodput with maximum aggregated packets

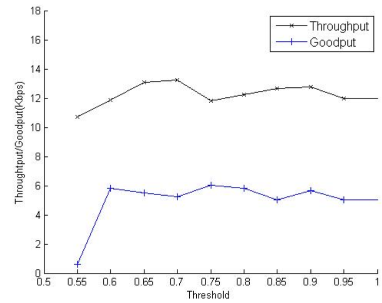

Fig. 11. Throughput and goodput with receiver similarity threshold

domain multicast unlike our approach discussed in this paper. Also, most WSN protocols are designed for low-power low-rate networks. They may not be directly applicable to IoT as human-interfaced devices and applications increase, requiring a higher rate and bandwidth.

For M2M and D2D communications, cluster-based multicast protocols have been developed [5], [15], [16], [27], [28]. However, most D2D or M2M works mainly focus on cellular networks to offload the base station. D2D and M2M protocols are generally constrained by the cellular network topology, which doesn't apply to most home or building applications. In contrast, our approach isn't constrained by any specific network topology. It is designed to enhance wireless network performance, only relying on basic information about link quality, link correlation, and receiver similarity.

VI. Conclusion

This paper presents a new high throughput multicast protocol for IoT communications via LC-aware forwarding and RI-ACK. Our simulation result shows that it outperforms CSMA in terms of throughput, goodput and RI-fairness. In the future, we will investigate more packet-effective approaches, in which, for example, our multicast protocol can be extended via network coding to reduce the aggregated packet length and more energy-efficient approaches in the multicast protocol design.

Acknowledgement

This work was supported, in part, by NSF grants CNS-1503590 and CNS-1526932.

References

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, *Internet of Things: A Survey on Enabling Technologies, Protocols and Applications.* Communications Surveys Tutorials, Issue: 99, 2015.

[2] L. Atzori, A. Iera, G. Morabito, *The Internet of Things: A survey .* Computer Networks, 2010

[3] A. Asadi, Q. Wang, V. Mancuso, *A survey on Device-to-device communication.* IEEE Communications Surveys and Tutorials, 2014.

[4] A. Castellani, N. Buit, P. Casari, M. Rossi, Z. Shelby, M. Zorzi, *Architecture and Protocols for the Internet of Things: A Case Study.* Pervasive Computing and Communications Workshops , 2010

[5] Y. Lin , Y. Hsu, *Multi-hop Cellular: A new architecture for wireless communications.* INFOCOM, 2000.

[6] R. Ananthapadmanabha, B. S. Manoj, C. S. Murthy, *Multi-hop cellular Networks: The architecture and Routing protocols.* Personal, Indoor and Mobile Radio Communications, 2001.

[7] B. Peng, T. Peng, Z. Liu, Y, Yang, C, Hu, *Cluster-based Multicast Transmission for device-to-device communication.* VTC, 2013

[8] N. Pratas, P. Popovski, *Low-Rate Machine-Type Communication visa Wireless D2d Links.* JSAC, 2013.

[9] J. Polastre, J. Hill, D. Culler, Versatile Low Power Media Access for Wireless Sensor Networks. SenSys, 2004

[10] I. Rhee, A. Warrier, M. Aia and J. Min, Z-MAC: a Hybrid MAC for Wireless Sensor Networks. IEEE/ACM Transactions on Networking, Vol. 16, 2008

[11] T. Zhu, Z. Zhong, T. He, Z. Zhang, *Explore link correlation for efficient flooding in wireless sensor network.* NSDI, 2010

[12] D. Couto, D. Aguayo, J. Bicket, R, Morris, A High-Throughput Path Metric for Multi-Hop Wireless Routing. Wireless Networks, 2005

[13] A. Basalamah, S. Kim, S. Guo, T. He, Y. Tobe, Link Correlation Aware Opportunistic Routing. IEEE Transactions on Wireless Communications, 2015

[14] S. Biswas, R. Morris, *ExOR: Opportunistic Multi-Hop Routing for Wireless Networks.* SIGCOMM, 2005.

[15] M. Chen, J. Wan, G. S., X. Liao, L. V.C.M., *A Survey of Recent Developments in Home M2M Networks.* IEEE Communications Surveys, 2014.

[16] Machine-to-Machine communications (M2M) Functional architecture. ETSI TS 102 690 V2.1.1, 2013

[17] S. Biswas, R. Morris, *Opportunistic Routing in Multi-Hop Wireless Networks.* SIGCOMM, 2004.

[18] RETF, *OSPF Version 2* 1998.

[19] D. Qiao S. Choi, K.G. Shin , *Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs,* IEEE Transactions On Mobile Computing, 2002.

[20] A. Pandya, B. Hamilton, A. Kansal, G. Pottie, Goodput and Delay in Networks with Controlled Mobility. Aerospace Conference, 2008

[21] S. Li., Y. R.W. , N. Cai, Linear network coding. IEEE Transactions on Information Theory, 2003

[22] K. Srinivasan, M. Jain, J. Choi, T. Azim, E. Kim, P. Levis, B. Krishnamachari, The κ Factor: Inferring Protocol Performance Using Inter-link Reception Correlation. Mobicom, 2010

[23] S. Chachulski, M. Jennings, S. Katti, D. Katabi, Trading Structure for Randomness in Wireless Opportunistic Routing. SIGCOMM, 2007

[24] M. Buettner, G. Yee, E. Anderson, R. Han, X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks. Sensys, 2006

[25] F. Ferrari, M. Zimmerling. L. Mottola, L. Thiele, Low-Power Wireless Bus. Sensys, 2012

[26] F. Ferrari, M. Zimmerling, L. Thiele, O. Saukh, Network Flooding and Time Synchronization with Glossy. IPSN, 2011

[27] K. Doppler, M. Rinne, C. Wijting, C. Ribeiro, K. Hugl, *Device to device communication as underlay to LTE-advanced networks.* IEEE Communications Magazine, 2009.

[28] J. Tan, S. Koo, *A survey of Technologies in Internet of Things.* IEEE International Conference on Distributed Computing in Sensor Systems, 2014.