

Research Article

Multihop Data Delivery Virtualization for Green Decentralized IoT

Lifeng Zhang,¹ Celimuge Wu,² Tsutomu Yoshinaga,² Xianfu Chen,³
Tutomu Murase,⁴ and Yusheng Ji⁵

¹The Youth College of Political Science, Inner Mongolia Normal University, 81 Zhawudalu Street, Saihan District, Hohhot 010022, China

²The Graduate School of Informatics and Engineering, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585, Japan

³The VTT Technical Research Centre of Finland, P.O. Box 1100, 90571 Oulu, Finland

⁴The Information Technology Center, Nagoya University, Chikusa-ku, Nagoya, Japan

⁵The Information Systems Architecture Research Division, National Institute of Informatics, 2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan

Correspondence should be addressed to Celimuge Wu; clmg@is.uec.ac.jp

Received 27 August 2017; Accepted 13 November 2017; Published 25 December 2017

Academic Editor: Jun Huang

Copyright © 2017 Lifeng Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Decentralized communication technologies (i.e., ad hoc networks) provide more opportunities for emerging wireless Internet of Things (IoT) due to the flexibility and expandability of distributed architecture. However, the performance degradation of wireless communications with the increase of the number of hops becomes the main obstacle in the development of decentralized wireless IoT systems. The main challenges come from the difficulty in designing a resource and energy efficient multihop communication protocol. Transmission control protocol (TCP), the most frequently used transport layer protocol for achieving reliable end-to-end communications, cannot achieve a satisfactory result in multihop wireless scenarios as it uses end-to-end acknowledgment which could not work well in a lossy scenario. In this paper, we propose a multihop data delivery virtualization approach which uses multiple one-hop reliable transmissions to perform multihop data transmissions. Since the proposed protocol utilizes hop-by-hop acknowledgment instead of end-to-end feedback, the congestion window size at each TCP sender node is not affected by the number of hops between the source node and the destination node. The proposed protocol can provide a significantly higher throughput and shorter transmission time as compared to the end-to-end approach. We conduct real-world experiments as well as computer simulations to show the performance gain from our proposed protocol.

1. Introduction

In recent years, Internet of Things (IoT) technologies have attracted great interests [1, 2]. With the increase of the number of IoT devices, it is difficult to provide all the communications using infrastructure-based wireless communications (including cellular networks and wireless access point-based communications) due to the cost of deploying and maintaining the infrastructures. Decentralized communication technologies can provide a low cost and efficient solution for emerging wireless Internet of Things (IoT) due

to the flexibility and expandability. However, the limited wireless resources pose new challenges on the design of an efficient end-to-end multihop transmission scheme.

Transmission control protocol (TCP) [3, 4] is a widely used transport layer protocol which provides reliable end-to-end data delivery. In order to avoid network congestion, TCP sender node adapts its congestion window size according to the reception status of data segments which are acknowledged by the destination node. The rationale behind this is that TCP assumes that the main reason for packet loss is due to the network congestion. However, this is not true in

wireless networks especially for multihop communications where the weak signal strength and random backoff incurred packet collisions are primarily responsible for packet losses.

In a multihop wireless communication, the end-to-end packet loss probability increases linearly with the packet hopping, which results in a small congestion windows size at the TCP sender (low throughput) and hence a large end-to-end delay. Thus, there is an urgent need to design an efficient protocol to handle the multihop data transmissions in an IoT environment.

There have been some studies discussing the transport layer design of multihop wireless networks [5–14]. Most of the existing approaches [5–9] focus on improving the congestion control of TCP. These approaches do not address the bandwidth underestimation of TCP in a multihop lossy environment. Since these protocols require modifications on TCP, the implementation difficulty is another problem. Some protocols have been proposed for specific IoT networks [10–14]. Depending on some specific protocols, these protocols lack generality. Existing studies also cover the efficient use of wireless resources for the purpose of green IoT networking [15–18]. All the existing protocols [5–18] do not discuss the performance drop problem of end-to-end retransmission in a multihop distributed wireless IoT environment adequately.

This paper focuses on efficient utilization of wireless resources in multihop wireless communications. For sensor data collection, multihop communications are usually required. At the same time, the real time data exchange between the source node and the destination node is not required. For example, a sensor node could upload large files such as video data to the cloud server for the surveillance purpose. For this kind of applications, the main objective is to successfully transfer large amount of data with small amount of wireless resources. These applications typically require high throughput but do not require the real time acknowledgment (response) from the receiver side (the destination node). In this paper, we propose a multihop data delivery virtualization approach which utilizes hop-by-hop acknowledgment instead of end-to-end feedback. The proposed protocol can achieve a higher throughput and shorter transmission time which is more energy efficient. The main contributions of this paper are as follows.

- (i) We propose a protocol which can efficiently solve the performance degradation problem of multihop TCP transmissions in a wireless IoT environment by virtualizing a multihop data transmission to multiple one-hop reliable transmissions. The protocol is easy to implement as it works on the top of TCP/IP protocol stack and does not require any modification on existing protocols.
- (ii) We conduct experiments with real devices and launch computer simulations to evaluate the protocol. Simulation results show that the proposed protocol can achieve a significant throughput improvement and delivery time reduction as compared to the conventional approach.

The remainder of the paper is organized as below. Section 2 gives a brief survey of related work. We give a detailed

description of the proposed protocol in Section 3 and perform a theoretical analysis in Section 4. Experimental results and simulation results are presented in Sections 5 and 6, respectively. Finally, we draw our conclusions and present future work in Section 7.

2. Related Work

There have been some studies discussing transport layer solutions for efficient data transmissions in multihop wireless networks. Some of them try to improve the congestion control of TCP, and some of them propose to design a new congestion control algorithm from scratch. Existing works also include some approaches which are specifically designed for green IoT networks.

2.1. TCP-Based Approaches. Talau et al. [5] have proposed an early congestion control approach which adjusts the size of congestion window according to the length of router queue. This approach is efficient in a congested scenario where the length of queue is large. However, the performance problem in a multihop lossy and less congested scenario is not discussed. Zhang et al. [6] have proposed a bandwidth delay product (BDP) estimation method and then designed a congestion control approach based on the BDP estimation. The BDP estimation approach solves the bandwidth overestimation problem by taking into account the contention delay at the MAC layer. In a multihop lossy network, the network bandwidth is underestimated due to the high probability of segment losses.

Schubert and Bambos [7] have proposed an approach which freezes the congestion window size when a deep fading happens in order to maintain a high sending rate at the sender side. Multihop transmissions are not addressed adequately in [7] as the congestion control is only conducted at the source node. Jude and Kuppaswami [8] have proposed a scheme in which the initial congestion window size is set based on the advertised window size. This is not suitable for a multihop TCP flow where the intermediate nodes do not advertise window size. Lee et al. [9] have proposed an approach which changes the slow start threshold in case of momentary link instability. However, it is difficult to know the reason of a segment loss especially for multihop communications. All these protocols [5–9] do not address the TCP performance drop problem in a multihop wireless IoT environment adequately.

2.2. Congestion Control Algorithms Specifically Designed for IoT. Li et al. [10] have proposed a congestion window adaptation scheme which employs a Q-learning algorithm based on network conditions. Since the observation of network condition is limited to delay variance, [10] cannot solve the problem of lossy wireless environment. Liaqat et al. [11] have proposed a social-similarity-aware congestion avoidance protocol which performs data transfer over TCP by taking advantage of similarity-matching social properties. Since it is difficult to get social relationship between nodes, [11] cannot provide a general solution. Govindan and Azad [12] have analyzed content delivery probability and delay in message queue telemetry transport protocol (MQTT) for wireless

sensor networks. However, [12] does not discuss how to improve the performance.

Study in [13] proposes an advanced congestion control algorithm for the constrained application protocol. The core concept of [13] is based on an accurate estimation of round trip time. Al-Kashoash et al. [14] have proposed GTCCF, a game theory-based congestion control framework for IEEE 802.15.4 networks. GTCCF takes into account node priorities and application priorities. These works [13, 14] do not discuss the small congestion window size problem in multihop lossy networks.

2.3. Green IoT Solutions. Arshad et al. [15] have provided a survey of green IoT technologies including energy efficient data centers, energy efficient transmission of sensor data, and energy efficient policies. Li et al. [16] have proposed an adaptive network coding scheme to improve the transmission efficiency in IoT networks. With dependence on software defined wireless networking technology, it is difficult to implement [16] in conventional networks. Liu and Ansari [17] have provided an architecture which utilizes an overlay spectrum sharing approach to facilitate device-to-device for IoT in heterogeneous cellular networks. Wali et al. [18] have proposed a Time-Spatial Randomization (TS-R) technique to mitigate the radio access network overload problem in a Long-Term Evolution-Advanced (LTE-A) cell. These existing studies [15–18] do not discuss the green IoT technologies in decentralized environments.

3. Proposed Protocol

3.1. Problem Definition and Protocol Concept. TCP is an end-to-end acknowledgment based data delivery approach. In TCP, the sender node adapts the sending rate by using a congestion control algorithm according to the acknowledgment packets successfully received (see Figure 1). The congestion control algorithm consists of two phases specifically the slow start phase and the congestion avoidance phase. In the slow start phase, the congestion windows size is initialized at 1 maximum segment size (MSS) and increases by 1 MSS upon reception of each acknowledgment (ACK) packet. In the congestion avoidance phase, upon the detection of a packet loss, the congestion windows size is reduced.

The throughput of a TCP connection is affected by the end-to-end packet loss probability. The end-to-end packet loss probability for N hop transmission can be calculated as

$$P_l = 1 - (1 - p_l)^N, \quad (1)$$

where p_l is the packet loss probability for each hop.

In order to show the effect of hop count on the TCP performance, here we analyze the TCP congestion window size which basically determines the throughput of a TCP connection. In the TCP slow start phase, the congestion window size is increased by 1 MSS upon reception of an ACK. When the end-to-end loss probability is P_l , the average

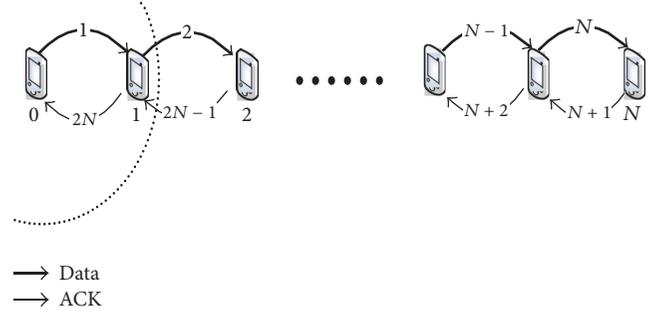


FIGURE 1: Data and acknowledgment in TCP (the numbers on the arrows show the transmission order of packets, and the dotted circle shows the transmission range of node 0).

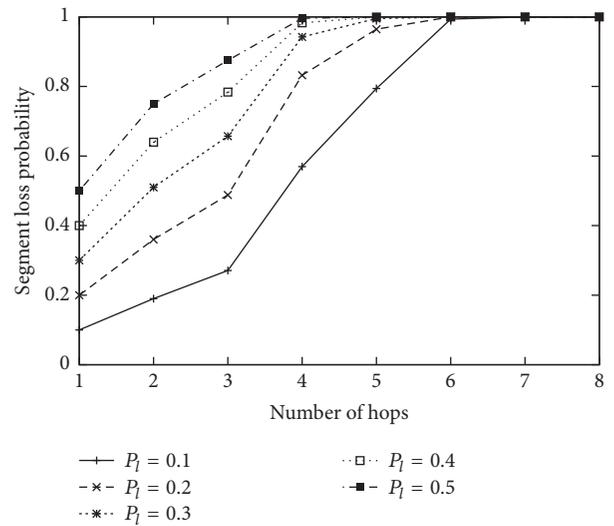


FIGURE 2: End-to-end segment loss probability for various numbers of hops.

congestion window in the slow start phase for the first 5 round trip times (RTTs) is

$$P_l + \sum_{i=1}^5 (2^{N_{rtt}-1}) (1 - P_l)^i \times (i + 1), \quad (2)$$

where N_{rtt} is the number (index) for RTT.

Figures 2 and 3 show the end-to-end segment loss probability and the corresponding congestion window for different numbers of hops. We can observe that the congestion window size of TCP drops drastically with the increase of hop count, resulting in low throughput for multihop lossy environment. This shows the problem of end-to-end retransmission approach that the hop count could drastically increase the possibility of packet loss. In order to solve this problem, we propose an efficient multihop data forwarding approach, namely, multihop data delivery virtualization, which virtualizes a multihop TCP communication to multiple one-hop reliable communications. With the virtualization, the proposed protocol enables hop-by-hop acknowledgment instead of end-to-end feedback and therefore utilizes the wireless resources better by avoiding underestimation of bandwidth.

TABLE 1: Implementation layer.

Application layer
Proposed protocol
Transport layer
Network layer
Data link layer
Physical layer

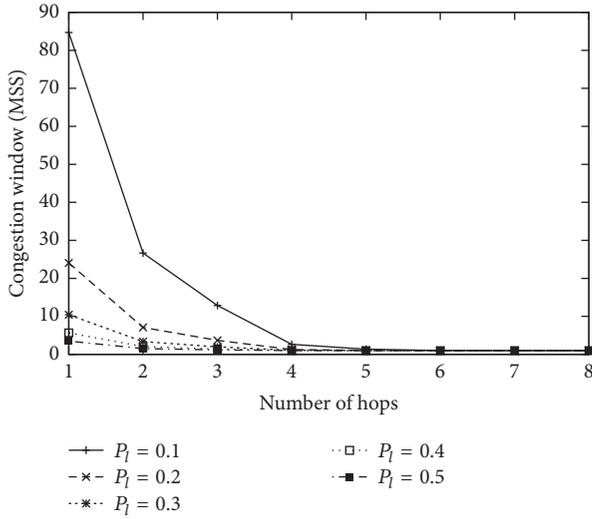


FIGURE 3: TCP congestion window size for various numbers of hops.

3.2. Multihop Data Delivery Virtualization. The proposed multihop data delivery virtualization approach is designed to solve the problem of TCP in a multihop lossy scenario where the segment loss probability increases with the number of hops. The proposed protocol works between the transport layer and the application layer (see Table 1). Therefore, any transport layer protocol (such as TCP and UDP) can be used. However, here we explain the implementation of the proposed protocol by using TCP as it is the most commonly used transport layer protocol in the Internet.

As shown in Figure 4, in the proposed approach, the data segments are acknowledged by using a hop-by-hop approach. The proposed approach works on the top of transport layer and uses multiple one-hop TCP connections to establish a multihop end-to-end connection. Since each one-hop communication is conducted based on TCP, the congestion window size at each TCP sender node is not affected by the number of hops between the source node and the destination node. This can facilitate an efficient use of wireless resources. The fairness among multiple flows is also ensured by each TCP sender node as the proposed protocol does not require any modification on TCP. As we observe from Figures 1 and 4, the proposed protocol does not increase the overhead in terms of ACK packets compared to the conventional multihop end-to-end TCP approach. Instead, the proposed protocol can reduce the data transmission overhead when a packet loss occurs by conducting retransmissions from an intermediate node while the ordinary approach conducts data retransmissions from the source node.

TABLE 2: Format of “SESS.INFO” message.

Session ID	Src IP	Dst IP
------------	--------	--------

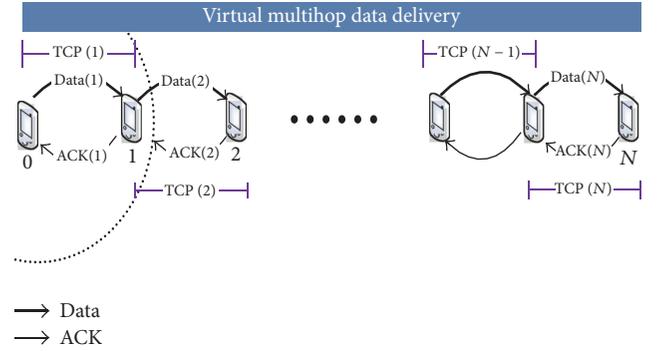


FIGURE 4: Multihop data delivery virtualization (numbers on the arrows show the transmission order of packets).

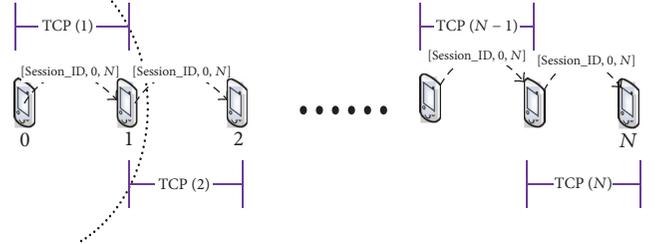


FIGURE 5: Multiple TCP sessions.

3.3. Protocol Implementation Details. In the proposed protocol, multihop data transmission session is maintained by a control message called “SESS.INFO” which contains the fields of session ID, source IP, and the destination IP (see Table 2). The session ID field is generated based on the flow identifier. As shown in Figure 5, the source node of each data flow transmits this information to the next hop node based on the routing information (the route to the destination can be acquired by any routing protocol such as [19, 20]). After reception of “SESS.INFO” message, each node maintains the session information locally. The corresponding information fields are shown in Table 3 where “Loc Pointer” denotes the pointer for the corresponding session data stored locally (a local buffer is used to store the unacknowledged data). The next hop node then sends the same information to the node downstream to the destination node. In this way, the session information is exchanged among all the forwarder nodes.

As shown in Figure 5, node 0 sends data to node 1, then node 1 forwards the data to its next hop node 2, and so on. Each data flow is a one-hop TCP transmission, and therefore a multihop data flow is virtualized into multiple one-hop data flows. As a result, the sending rate (congestion window size for TCP) at each sender node is improved, which means better utilization of wireless resources environment.

Algorithms 1 and 2 show the actions at each source node and forwarder node, respectively. Upon reception of the first

```

Check the corresponding session information from the local
storage.
Contact the routing agent to find the next hop node.
if (the corresponding session information is not exchanged
among the forwarder node) then
    Send "SESS_INFO" message to the next hop node.
    Update local session information.
end if
Send the data to the next hop node.

```

ALGORITHM 1: Actions at each source node upon reception of a packet from the application layer.

```

if (the packet is a "SESS_INFO" message) then
    Create a route the sender of this packet (the route is used
to send TCP ACK).
    Update local session information.
else
    Contact the routing agent to find the next hop node.
    Send the data to the next hop node.
    Send ACK to the sender of this packet.
end if

```

ALGORITHM 2: Actions at each forwarder node upon reception of a packet.

TABLE 3: Session information maintained locally.

Session ID	Src IP	Dst IP	Loc Pointer	Timestamp
------------	--------	--------	-------------	-----------

packet of a certain flow, the source node sends the corresponding session information, specifically "SESS_INFO" message, to the next hop node. The "SESS_INFO" message will be forwarded further until reaching the destination node. Each forwarder node maintains the session information about each data flow locally and acknowledges the data packets received based on the corresponding session information.

3.4. Cooperation with Routing Agent. Due to the topology changes, there could be some route breaks. The handling of route error should be done by the routing agent. However, the proposed protocol reports the link failure event to the routing agent in order to eliminate the effect of route failure on the network performance. As shown in Algorithm 3, when a link failure happens, the event will be reported to the routing agent to optimize the end-to-end route. The node that sensed the link failure transmits the data in the local buffer by establishing a new route to the destination.

4. Theoretical Analysis

Here we analyze the performance of the proposed protocol for delay tolerant applications such as sensor data collections. In the proposed protocol, since the traffic source node does not require the acknowledgment of a data packet from the

```

Report the link failure event to the routing agent.
if (the local buffer is not empty) then
    Contact the routing agent to create a new route to the
destination.
    Send the data in the buffer to the destination using the
route.
end if

```

ALGORITHM 3: Actions when a link failure happens.

destination node, the multihop transmission can be separated into multiple one-hop transmissions similar to the delay tolerant approach in [21]. Therefore, the proposed protocol can reduce the number of concurrent transmission nodes, resulting in more efficient wireless resource utilization.

4.1. Probability of Collision. In the IEEE 802.11p standard, the backoff time is a random number which is drawn from a uniform distribution over the interval $[0, CW]$ where CW is the current contention window. If two sender nodes are located closer than the sensing range and they choose the same contention window size, there will be collisions at some receiver nodes. Since a transmission is successful only when all sender nodes choose different backoff values, we can calculate the probability of collisions as

$$P_c(N) = \begin{cases} 1, & \text{if } CW + 1 \leq N \\ 1 - \frac{\prod_{k=0}^{N-1} (CW + 1 - k)}{(CW + 1)^N}, & \text{otherwise,} \end{cases} \quad (3)$$

where N is the number of sender nodes. In a multihop communication scenario, if N_f is the number of traffic flows, then the collision probability is $\bar{P}_c = P_c(2N_f)$. In the proposed protocol, the probability is $\hat{P}_c = P_c(N_f)$. As shown in Figure 6, the proposed protocol can attain lower collision probability by reducing the number of sender nodes.

4.2. Probability of End-to-End Packet Loss. A packet can be lost due to either collisions or weak signal strength. The packet loss probability for a 1-hop transmission can be calculated as

$$P(N) = P_c(N) + P_l(N) - P_c(N) \times P_l(N), \quad (4)$$

where $P_l(N)$ is the packet loss probability due to weak signal strength. For H -hop transmission, the end-to-end loss probability is

$$P_{e2e}(N, H) = 1 - (1 - P(N))^H, \quad (5)$$

where H is the number of hops between two communication nodes. Based on (5), we show the TCP segment loss probability for various numbers of hops in Figure 7. Since the proposed approach can reduce the probability of collisions and can avoid using multihop communications, the probability of TCP segment loss can be reduced significantly.

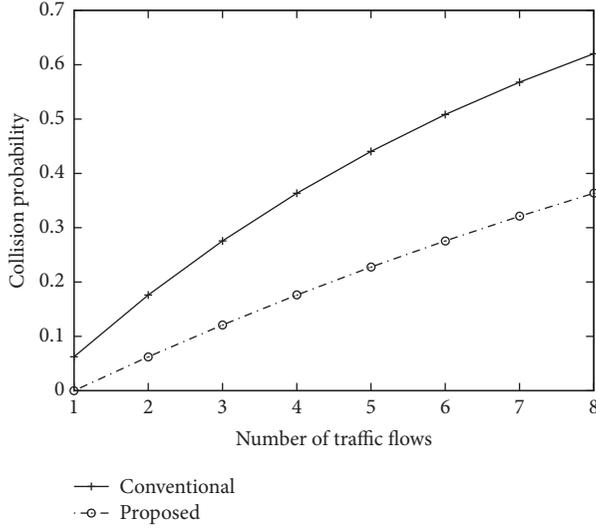


FIGURE 6: Collision probability for different numbers of traffic flows.

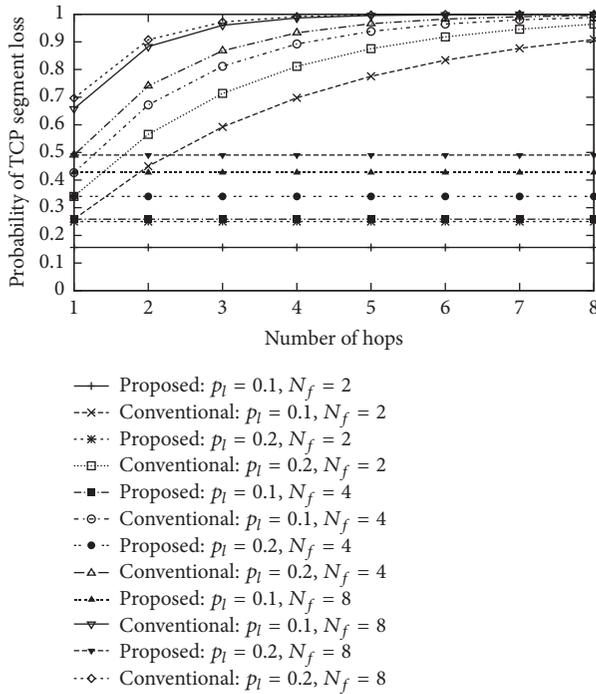


FIGURE 7: TCP segment loss probability for different numbers of hops.

5. Experimental Results

We conducted real-world experiments using devices as shown in Figure 8. Each node is equipped with IEEE 802.11g radio interface. In order to show the effect of the proposed protocol more clearly, we used a chain topology where all the devices are deployed on a straight line. The number of hops from the source node to the destination node was 2. The proposed protocol was compared with the conventional TCP

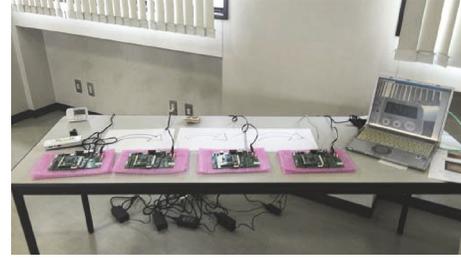


FIGURE 8: Devices used for the experiment.

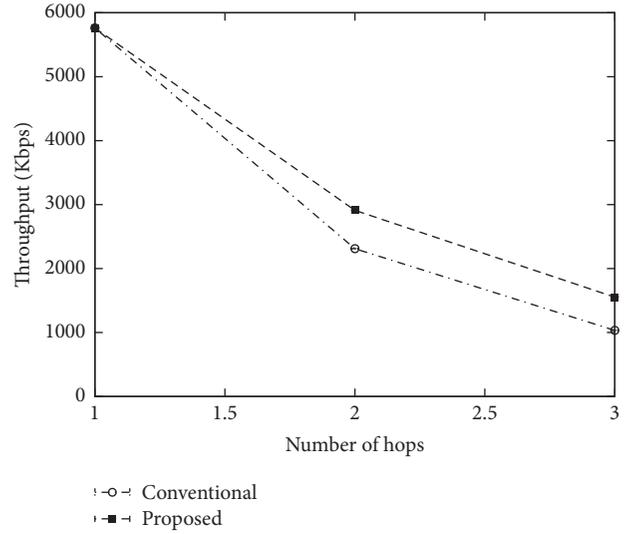


FIGURE 9: Throughput for different numbers of hops.

which uses end-to-end acknowledgment to set the congestion window size.

Figure 9 shows the TCP throughput for different internode distances. We can observe that the proposed protocol can achieve up to 50% throughput improvement as compared to the conventional end-to-end acknowledgment approach of TCP. This is because the proposed protocol can utilize the wireless resources better by using hop-by-hop acknowledgment which could result in higher sending rate (larger congestion window size) at each node.

6. Simulation Results

In order to clearly explain the performance of the proposed protocol, we conducted computer simulations using QualNet 6.1 [22]. Specific simulation parameters are shown in Table 4. Other parameters were the default settings of QualNet 6.1. The proposed protocol was compared with the conventional end-to-end based retransmission approach (multihop TCP). In the following simulation results, the error bars indicate the 95% confidence intervals.

6.1. Effect of Various Internode Distances. The number of hops from the source node to the destination node was 2. Figure 10 shows the throughput for various internode distances. The

TABLE 4: Simulation environment.

Simulator	QualNet 6.1
Topology	Chain topology
PHY	IEEE 802.11g
MAC	54 Mbps (auto rate fallback)
Transmission range	Around 420 m
Packet size	512 bytes
TCP variant	New Reno
Simulation time	500 s

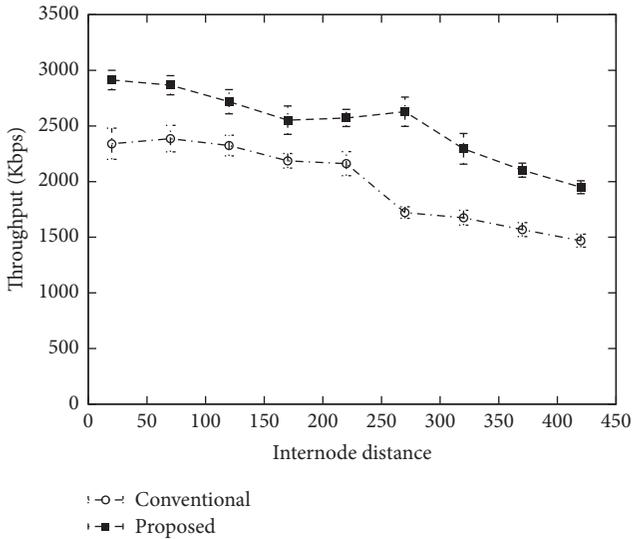


FIGURE 10: TCP throughput for internode distances (2-hop).

proposed protocol can achieve larger throughput than the conventional end-to-end based retransmission approach in all scenarios. By using the multihop data delivery virtualization approach, the proposed protocol shows up to 50% throughput improvement over TCP, which can be expected to provide significantly higher quality of service to end users. Figure 11 shows the required time for sending 10 MB data. The proposed protocol can efficiently reduce the required active time (including transmission time, contention time, and all other overheads), which is more energy and resource efficient.

6.2. *Effect of Various Numbers of Hops.* We evaluated the effect of hop count in different internode distances. Figures 12, 13, and 14 show the throughput for various numbers of hops in short, medium, and long internode distances correspondingly.

The simulation results show that the throughput improvement of the proposed protocol is dependent on both the internode distance and the number of hops. The advantage of the proposed protocol becomes more notable with the increase of the number of hops. As the number of hops increases, the end-to-end segment loss probability increases, resulting in a smaller congestion window size. This is the

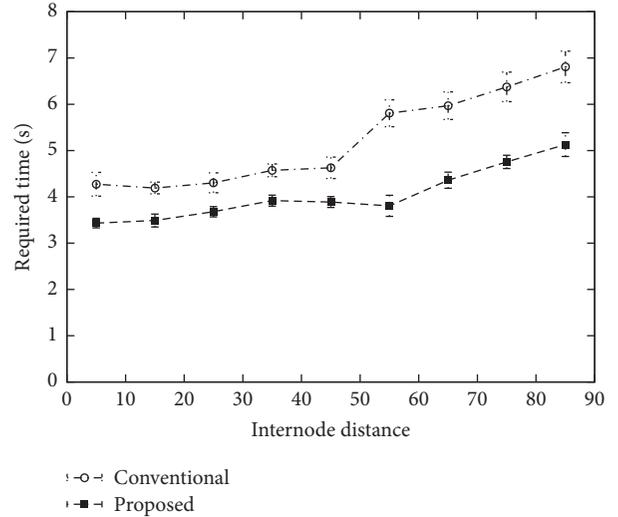


FIGURE 11: Required time for sending 10 MB data.

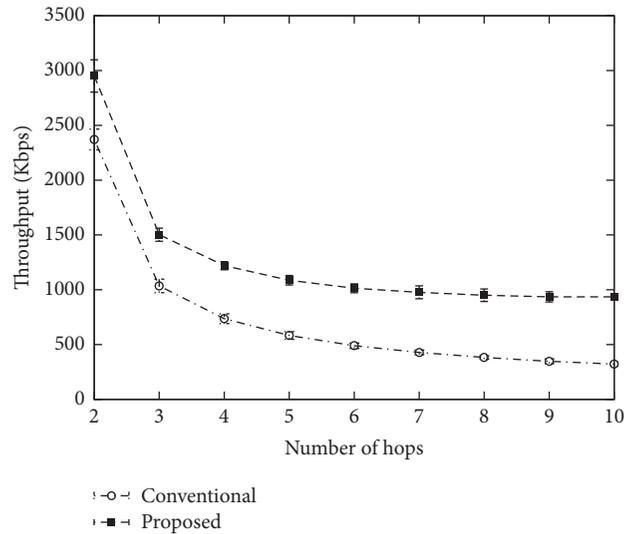


FIGURE 12: Throughput for various numbers of hops (short internode distance: 20 m).

main reason for the throughput degradation of the conventional approach.

The performance gain of the proposed protocol is also dependent on the internode distance. When the number of hops is small (in the case of 2), the lower the packet loss probability of a link is, the larger the gain is. However, when the number of hops is large (in the case of 10), the proposed protocol shows the highest gain in the short internode distance case. This is because a larger internode distance would incur a higher delay at each hop due to the retransmissions which affect the overall throughput in general.

Figures 15, 16, and 17 show the required time for sending 10 MB data in short, medium, and long internode distances correspondingly. We can observe that the proposed approach can shorten the time by up to 60%. This improvement is significant especially for battery-powered wireless sensor

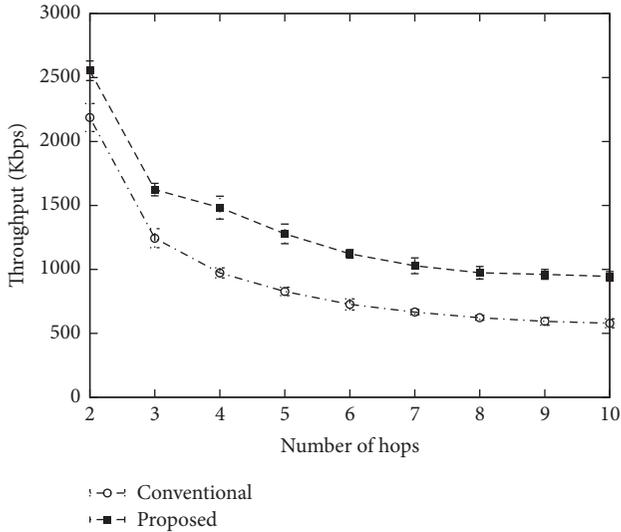


FIGURE 13: Throughput for various numbers of hops (medium internode distance: 170 m).

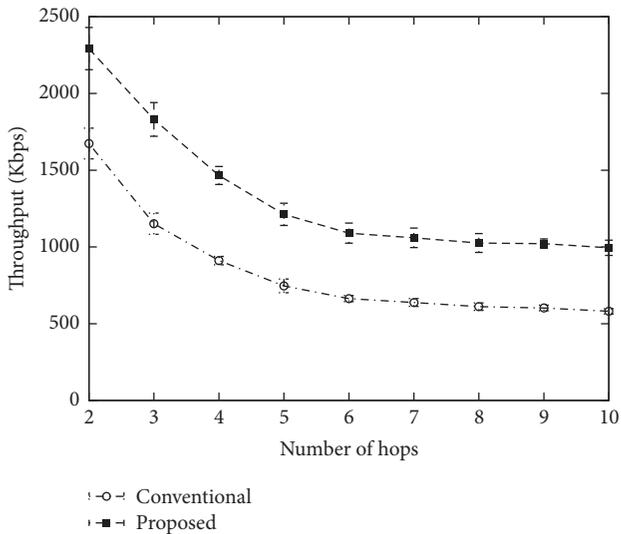


FIGURE 14: Throughput for various numbers of hops (long internode distance: 320 m).

networks as our protocol could enable more efficient duty cycling.

7. Conclusions

We proposed a multihop data delivery protocol for the distributed IoT environment. The proposed protocol uses multiple one-hop data flows to virtualize a multihop end-to-end data transmission. By using hop-by-hop acknowledgment, the proposed protocol is able to provide higher throughput as compared to the conventional end-to-end acknowledgment approach which is used by TCP. We conducted theoretical analysis, real-world experiment, and computer simulations to evaluate the performance of the proposed protocol and showed the advantage of the proposed protocol over the

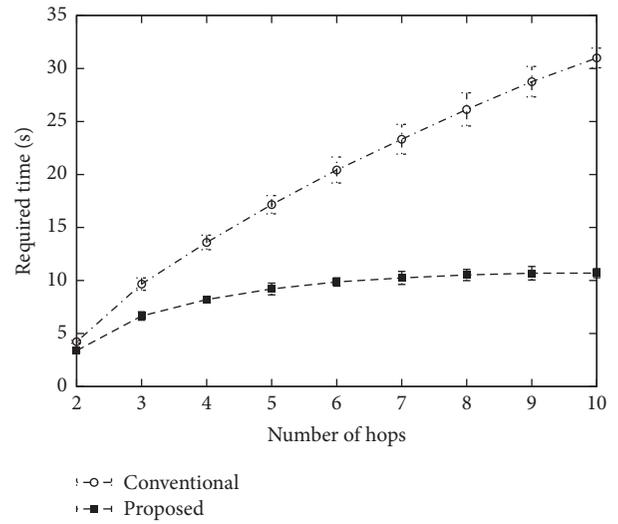


FIGURE 15: Required time for sending 10 MB data (short internode distance: 20 m).

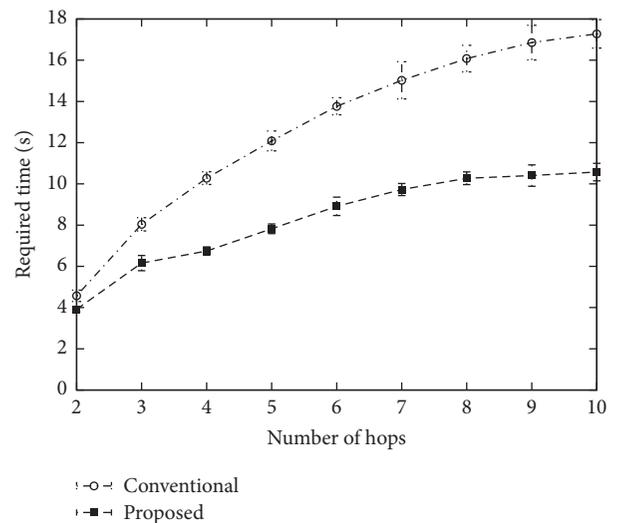


FIGURE 16: Required time for sending 10 MB data (short internode distance: 170 m).

conventional approach. Evaluation results show that the proposed protocol can achieve up to 50% throughput improvement over the end-to-end approach in 3-hop scenarios. By shortening the required transmission time at each node, the proposed protocol is able to provide a more energy and cost efficient solution for data transmissions in a multihop decentralized IoT environment.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

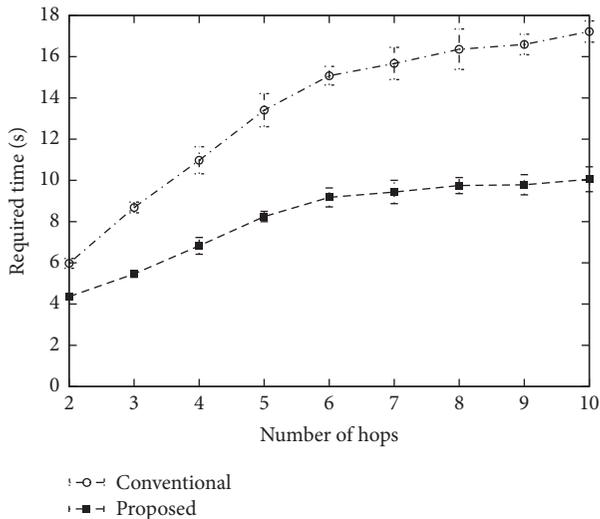


FIGURE 17: Required time for sending 10 MB data (long internode distance: 320 m).

Acknowledgments

This work was partially supported by NII Joint Research Funding, JSPS KAKENHI Grant nos. 16K00121 and 16H02817.

References

- [1] K. Wang, Y. Wang, Y. Sun, S. Guo, and J. Wu, "Green Industrial Internet of Things Architecture: An Energy-Efficient Perspective," *IEEE Communications Magazine*, vol. 54, no. 11, pp. 48–54, 2016.
- [2] K. Wang, X. Qi, L. Shu, D.-J. Deng, and J. J. P. C. Rodrigues, "Toward trustworthy crowdsourcing in the social internet of things," *IEEE Wireless Communications Magazine*, vol. 23, no. 5, pp. 30–36, 2016.
- [3] J. Luo, J. Jin, and F. Shan, "Standardization of Low-Latency TCP with Explicit Congestion Notification: A Survey," *IEEE Internet Computing*, vol. 21, no. 1, pp. 49–55, 2017.
- [4] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock, "Host-to-host congestion control for TCP," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 3, pp. 304–342, 2010.
- [5] M. Talau, M. Fonseca, A. Munaretto, and E. C. G. Wille, "Early congestion control: A new approach to improve the performance of TCP in ad hoc networks," in *Proceedings of the 2016 7th International Conference on the Network of the Future, NOF 2016*, November 2016.
- [6] X. Zhang, B. Yang, and Y. Ma, "TCP congestion control based on accurate bandwidth-delay product in wireless Ad hoc networks," in *Proceedings of the 2014 23rd International Conference on Computer Communication and Networks, ICCCN 2014*, August 2014.
- [7] K. Schubert and N. Bambos, "Wireless power controlled TCP with holdover," in *Proceedings of the IEEE ICC*, pp. 1–6, 2016.
- [8] M. J. A. Jude and S. Kuppaswami, "Enhanced window increment and adaptive recovery TCP for multi-hop wireless networks," *IEEE Electronics Letters*, vol. 53, no. 6, pp. 438–440, 2017.
- [9] J. Y. Lee, H. Kim, W. Lee, and H. Kim, "Adaptive Transmission Scheme for TCP in Wireless Multi-Hop Network," in *Proceedings of the 41st IEEE Conference on Local Computer Networks, LCN 2016*, pp. 503–506, UAE, November 2016.
- [10] W. Li, F. Zhou, W. Meleis, and K. Chowdhury, "Learning-based and data-driven TCP design for memory-constrained IoT," in *Proceedings of the 12th Annual International Conference on Distributed Computing in Sensor Systems, DCOSS 2016*, pp. 199–205, May 2016.
- [11] H. B. Liaqat, F. Xia, J. Ma, L. T. Yang, A. M. Ahmed, and N. Y. Asabere, "Social-similarity-aware TCP with collision avoidance in ad hoc social networks," *IEEE Systems Journal*, vol. 99, pp. 1–12, 2014.
- [12] K. Govindan and A. P. Azad, "End-to-end service assurance in IoT MQTT-SN," in *Proceedings of the 2015 12th Annual IEEE Consumer Communications and Networking Conference, CCNC 2015*, pp. 290–296, January 2015.
- [13] A. Betzler, C. Gomez, I. Demirkol, and J. Paradells, "CoAP congestion control for the internet of things," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 154–160, 2016.
- [14] H. A. Al-Kashoash, M. Hafeez, and A. H. Kemp, "Congestion Control for 6LoWPAN Networks: A Game Theoretic Framework," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 760–771, 2017.
- [15] R. Arshad, S. Zahoor, M. A. Shah, A. Wahid, and H. Yu, "Green IoT: An Investigation on Energy Saving Practices for 2020 and Beyond," *IEEE Access*, vol. 5, pp. 15667–15681, 2017.
- [16] J. Li, Y. Liu, Z. Zhang, J. Ren, and N. Zhao, "Towards Green IoT Networking: Performance Optimization of Network Coding Based Communication and Reliable Storage," *IEEE Access*, vol. 5, pp. 8780–8791, 2017.
- [17] X. Liu and N. Ansari, "Green Relay Assisted D2D Communications With Dual Batteries in Heterogeneous Cellular Networks for IoT," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1707–1715, 2017.
- [18] P. K. Wali, A. A. N, and D. Das, "Optimal Time-Spatial Randomization Techniques for Energy Efficient IoT Access in LTE-Advanced," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 7346–7359, 2017.
- [19] C. Wu, Y. Ji, F. Liu, S. Ohzahata, and T. Kato, "Toward practical and intelligent routing in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 12, pp. 5503–5519, 2015.
- [20] C. Wu, X. Chen, Y. Ji et al., "Packet size-aware broadcasting in VANETs with fuzzy logic and RL-based parameter adaptation," *IEEE Access*, vol. 3, pp. 2481–2491, 2015.
- [21] C. Wu, Y. Ji, S. Ohzahata, and T. Kato, "Can DTN improve the performance of vehicle-to-roadside communication?" in *Proceedings of the 26th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications, PIMRC 2015*, pp. 1934–1939, September 2015.
- [22] QualNet, 2017, <http://web.scalable-networks.com/content/qualnet>.



Hindawi

Submit your manuscripts at
<https://www.hindawi.com>

