Priority-Aware Bulk Data Transfer in Low-power IoT Networks

Wenliang Mao[1], Zhe Wang[1], Zhiwei Zhao[1,*] and Geyong Min[2]
[1] College of Computer Science and Engineering, University of Electronic Science and Technology of China
[2] College of Engineering, Mathematics and Physical Sciences, University of Exeter
{wenliang,wangzhe}@mobinets.org; zzw@uestc.edu.cn; g.min@exeter.ac.uk

Abstract

With the remarkable development of the Internet of Things (IoT), the computation demands from IoT networks are explosively increasing. Due to the lack of computing resources, edge computing has recently become a good choice for enhancing the IoT performance, where the IoT devices can offload their tasks to the edge server nodes. In this framework, task offloading directly impacts the latency of IoT tasks. Bulk data transfer is one of the most important modules in task offloading, of which the performance can be easily degraded by the unreliable wireless links of the IoT devices. However, the existing works either under-utilize the spectrum or cannot guarantee the multi-task priorities. To achieve a good trade-off between the transmission efficiency and multi-task priorities, in this paper, we propose a priority-guaranteed and rapid approach for bulk data transfer. By considering link qualities and priorities in the organization of the transfer buffer, we can 1) fully exploit the wireless spectrum and 2) strictly guarantee the multi-task priorities. We conduct simulation experiments and compared the performance with the state-of-the-art works. The results show that our work improves the transmission efficiency while guaranteeing the multi-task priorities.

1 Introduction

In recent years, the development of microelectronic and low-power wireless technologies has enabled the rapid development and scale-up of Internet-of-Things (IoT) networks [4]. Specifically, as the devices and applications are explosively increasing, it becomes more and more challenging to meet the computation demands of the IoT networks [12, 32]. To deal with this problem, edge computing emerges as a promising alternative to provide additional computational resources for IoT networks [19,21]. In the Edge-for-IoT framework, the IoT devices usually are lower-powered and energy-constrained, which becomes a restriction for improving the performance of IoT network. With edge severs deployed in the IoT network, there exists an alternative method that the IoT devices can offload their tasks to the edge computing servers, such that the task efficiency can be improved and the energy of IoT devices can be saved [1, 30, 33]. Task offloading, which plays an essential role in edge computing for IoT, highly relies on the efficiency of wireless task transfer. For the low-power IoT devices, task transfer is even more important as its performance can be degraded by the unreliable wireless links of the low power radios [14, 34]. For example, the popular IoT platform TelosB is equipped with the low-power radio CC2420 [16] [28]. Its max radio power is 0dBm and can be easily interfered by the environmental noise and inference.

Considering the increasing application demands, each IoT device tend to have multiple tasks to offload and thus requires transferring bulk data of these tasks to the edge servers. There are some existing works on bulk data transfer that can be employed in task offloading [7, 17, 24]. In these works, the data object is divided into small pages, each of which consists of a number of packets. Meanwhile, time is sliced into slots and the slot length is just enough be used to transfer one page. Each device operates in three modes:transmission (T slot), reception (R slot) and sleep (S slot), and each mode only continues for one time slot [7, 17, 24]. One of the major limitations for these works is the time slots cannot be fully utilized when there are retransmissions (analyzed in Section 2 on lossy links. To deal with problem, Zhao et al. [31] proposed out-of-order transmissions for bulk data transfer (ULTRA). By inserting as many packets as possible in each page, the time slots can be fully utilized.

While this approach achieves good performance for bulk data transfer in general cases, it cannot well support the transfer of multiple tasks. The reason is that IoT tasks often have strict priorities [15] which require the high-priority tasks to be delivered earlier than the low-priority tasks. ULTRA, although achieving good overall performance, cannot guarantee such priorities due to its out-of-order transmissions. We can also consider transferring these tasks separately, however, the overall transmission performance will be largely degraded due to the fragmented packet organizations. As a result, there exists a tradeoff between the overall transmission efficiency and priority guarantee. To achieve a good tradeoff, we propose a fast

International Conference on Embedded Wireless Systems and Networks (EWSN) 2019
25–27 February, Beijing, China

ISBN: 978-0-9949886-3-8

and priority-guaranteed approach of bulk data transfer for task offloading in Edge-for-IoT networks. 1) By organizing pages according to the priorities of tasks and the packet loss rate, we guarantee the task priorities. 2) By fulfilling every transmission slot as full as possible, we improve overall transmission performance (under the guarantee of priorities) Compared to the existing works, our approach can find a good tradeoff between priorities and overall transmission efficiency. We conduct simulation experiments and the results show that the proposed work improves the transfer efficiency compared to the conventional schemes, while guaranteeing the task priorities. The major contributions of this paper are summarized as follows.

- We propose a novel transmission scheme for bulk data transfer which considers both transmission efficiency and taks priorities.
- Based on the scheme, we propose a protocol which organizes pages during the task transfer, guaranteeing the task priorities.
- We conduct simulation experiments to study the performance of the proposed scheme. The results show that the scheme achieves fewer transmission rounds and guarantees task priorities.

The rest of the paper is organized as follows. Section 2 reviews the related works. Section 3 presents the main design of the proposed work. The simulations and evaluation analysis are given in Section 4. Section 5 concludes this work.

2 Related Work

Bulk data transfer has been one of the most important problems in low-power wireless networks and has attracted increasing research attention as the paradigm of edge computing emerges recently. There are a large number of existing studies about bulk data transfer, which can be roughly divided into two categories: structureless protocols and structured protocols.

Structureless protocols [3, 8, 9, 26, 29] include Deluge, MNP and ECD. These protocols implement reliable transmission mainly through employing three-way handshake and NACKs [2, 11, 13]. By dividing a data object into small data pages, the multi-hop pipelining is enabled. Deluge is the default transfer protocol in TinyOS [10], which transfers the data objects page by page. Deluge randomly selects forwarders, which can bring broadcast storm problem [25]. For improving Deluge, both MNP and ECD propose sender selection algorithms to avoid broadcast storm problem. Furthermore, When a node fails in the sender contention phase, MNP will turn off the node's radio for a better performance, which is called sleep scheduling. ECD reaches a better performance through adjusting the packet size dynamically. There are also other schemes which employ rateless coding to enhance the transfer efficiency [5,6,20,27]. However, structureless protocols heavily rely on the quality of links, so that they are not as suitable for dense network as structured protocols.

Structured protocols: In structured protocols, the network is formed in structures which consists two kinds of network nodes: core nodes and non-core nodes. After the topology structure is established, the sink first transfers the data object to all the core nodes, then each core node transfers the data object to all its neighboring non-core nodes. One of the structured protocols Sprinkler [18], requires geography information and then establishes a minimum connected dominating set (MCDS), which reduces transferring overhead by minimizing the number of core nodes. CoCo [23], is a recent structured protocol established on the sleep scheduling considering link correlation [22, 35]. The advantage of CoCo is that CoCo comprehensively considers the link characteristics before the topology structure is established. Another recent study aims at reducing the retransmissions, called ULTRA [31]. In ULTRA, a sender always tries to send packets as many as possible to its child nodes. By using disordered packets from multiple data objects to fulfill the transmission slots continuously, the data object can be pumped into the network as soon as possible.

Although ULTRA has an outstanding performance in the aspect of reducing transmission rounds, it neglects the object priorities. By simply fulfilling the unfull transmission slots, UlTRA may interrupt the task priorities, where the high priority objects can be delivered later than those with low priorities. In this case, the low priority objects have taken over some transmission rounds of high priority data object accidentally, thus the latency of high priority data objects may exceeds the deadline. This shortcoming of ULTRA motivates us to combine the task priorities and transmission efficiency into the bulk data transfer.

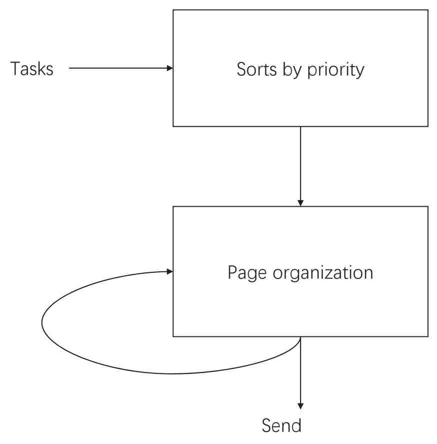


Figure 1. The main framework.

3 Main Design

In this section, we present our main design. We first introduce the framework of our transmission scheme, and then present the page organization details.

3.1 Overview

In our proposed scheme, as we can see in Figure 1, firstly, the sender sorts all the tasks needed to be offloaded according to the task priorities, and then divides those tasks into pages. In every transfer slots (T-slot), the sender only transfers one page. After receiving the page, the receiver will reply the sender a REQ which includes a bitmap indicating the reception situation of every packets in the page. Until

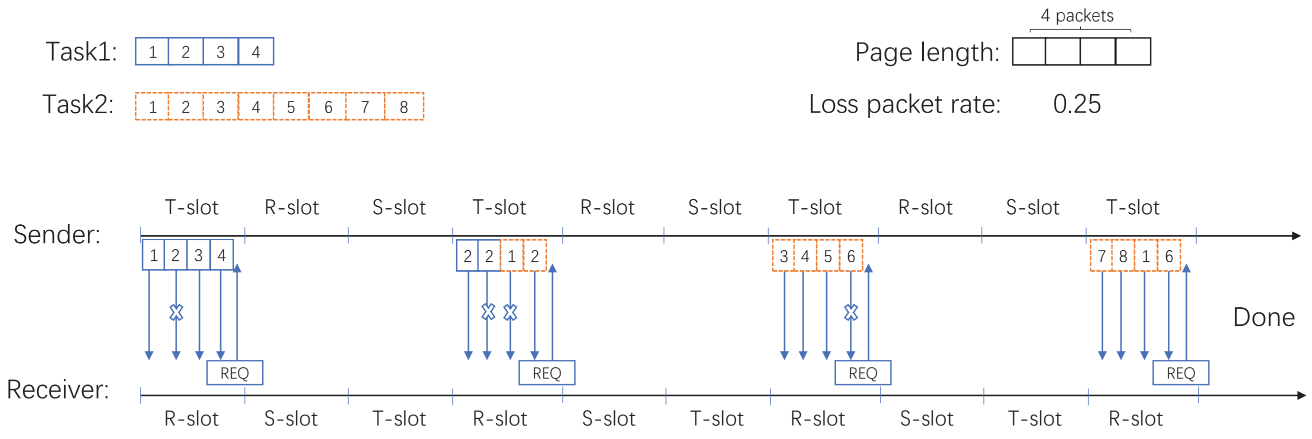


Figure 2. Illustration of our transfer scheme.

the T-slot to transfer the last page of the task, the sender fulfill the following pages by using the packets needed to be retransferred according to the previous REQs and the priority of the task.

Figure 2 illustrates our transfer scheme briefly. As we can see, Task 1 consists of 4 packets with a high priority, while task 2 consists of 8 packets with a low priority. In the first transmission round, the sender sends one page which consists of 4 packets to the receiver. However, the packet #2 has lost unfortunately. Considering the priority of task 1 and the packet loss rate, the sender sends the page which consists of two packets (#2) and two packets of task 2, in the second transmission round. Meanwhile, we can notice that although one of the two #2 packets of task 1 is lost again, task 1 has been still offloaded successfully. Then in the last transmission round, because of the low priority of task 2, the sender just simply retransferred the lost packets, instead of adding extra packets of those loss into the page. The above process shows how we deal with the high-priority tasks in the proposed scheme.

3.2 Page Organization

In this section, we will introduce how the data pages in our transfer scheme are organized. Then we present the algorithm of organizing a page.

Each page generally consists of several packets. The number of packets in per page is constant due to that the length of transmission slots is constant, When the sender needs to organize a page for sending, there are three different cases as follows. 1) The number of packets remaining in the current task is larger than the page size; 2) The number of packets remaining in the current task is smaller than the page size, but some packets of current task need to be retransferred; 3) Among the remaining packets, some need to be retransferred and the other packets are from the next task. For different cases, there are different methods of organizing pages. In the first case, we sequentially choose the packets of the current task to fulfill the slot. In the second case, we choose all of the remaining packets to constitute the front part of the slot, and then fulfill the remaining slot by the packets to be retransferred. In the third case, we choose the packets to be retransferred to constitute the page. If there are not enough packets for an entire slot, we add the same packets several times in the slot according to the packet loss rate. If that still cannot fulfill the slot, we choose the packets from the next task to fulfill that slot.

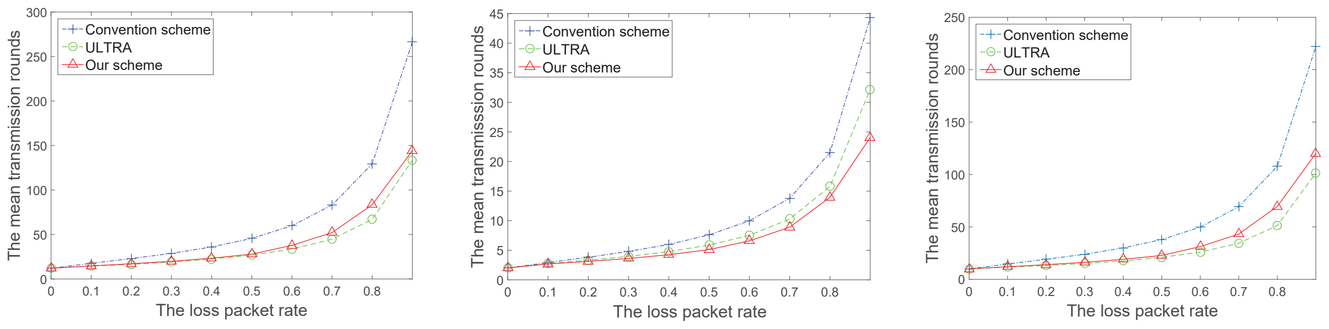
Algorithm 1 shows the details about the page organization. The input is the packet number of task t, n_t, the packet number of per page n_p, the packets set of task t, P_t, the packet loss rate r, the priority of current task p. The output is the packet set P_p of the current slot. This Algorithm can be divided into two parts: the first part simply uses the task packets to fulfill slots sequentially until the rest packets of task cannot fulfill a slot; the second part uses the retransfer packets and the next task packets to fulfill the transmission slot according to our strategy.

First part(Line 1-4): First of all, We figure out how many full pages can the task be divided into, which is n. n is determined by the task packets number n_t and the packets number of per page n_p, through $n = floor(n_t/n_p)$.(Line 1) After that, algorithm just simply divides the front $n{*}n_p$ packets of current task into n pages. and then outputs and sends those n pages.

Second part(line 5-19): We first add the remaining packets of the current task into the slot, which are not enough yet to fulfill a transmission slot. After that, if the task is in a high priority and there are packets needed to be retransferred according to the REQs, we add each packet-to-retransmit into the slot by f times(Line 9), until all the retransferred packets are retransferred successfully. f is determined by the packet loss rate r, through $f = ceil(1/(1-r))$ (Line 7). Until there are not retransferred packets and the slot is unfull, we add the packets of next task into the slot (Line 11). If the task is with a low priority and there are packets need to be retransferred, we add these packets-to-retransmit into the slot (Line 16). In the same way, until there are not any packets to be retransferred, we add the packets of next task into the slot (Line 18).

4 Simulation

In this section, a simulation experiment is provided to verify the validity of our transfer scheme. We first introduce



(a) Total two tasks (b) Task 1 (c) Task 2

Figure 3. The mean transfer times of tasks in different transfer schemes

```
Algorithm 1: The Page Organization Algorithm
Input  : Task packets number $n_t$, packets number of per page $n_p$, task
         packets set $P_t$, loss packet rate $r$, the priority of current task $p$
Output: Packets set $P_p$ of one page
1  $n = floor(n_t/n_p)$ // Round down to get the page number
2  for $i = 0; i < n; i++$ do
3      $P_p \Leftarrow P_t[i * n_p, (i+1) * n_p]$;
4      return Send $P_p$ and clear $P_p$
5  while $(P_t \neq \phi)$ and there are not packet needed be retransferred do
6      $P_p \Leftarrow$ rest of $P_t$ if $p ==$ high then
7          $f = ceil(1/(1-r))$ //Retransfer a same packet $f$ times
           while $P_p$ not full do
8              if There are packets need to be retransferred then
9                  $P_p \Leftarrow f*$ a retransfer packet // add $f$ same
                   retransfer packets
10             else
11                 $P_p \Leftarrow$ a packet of next packet
12         return Send $P_p$ and clear $P_p$
13     if $p ==$ low then
14         while $P_p$ not full do
15             if There are packets need to be retransferred then
16                 $P_p \Leftarrow$ a retransfer packet
17             else
18                 $P_p \Leftarrow$ a packet of next packet
19         return Send $P_p$ and clear $P_p$
```

the simulation setup, then present and analyze the simulation results.

4.1 Simulation Setup

We assume that the sender has two tasks to offload. Task 1 consists of 10 packets with a high priority. Task 2 consists of 50 packets with a low priority. Each slot contains five packets. Our transfer scheme is compared with two different transfer schemes: strictly sequential scheme (which transmits the packets of the next page until the current page is fully delivered) and ULTRA [31] transfer scheme.

- The strictly sequential scheme: The sender sends data objects sequentially in pages, and when receiving a REQ, the sender retransfers the lost packets at the next slot. The sender will not send packets of the next page until the current page is fully delivered.
- ULTRA transfer scheme: The sender sends tasks following the page sequences. When a page is unfull, the sender uses packets-to-retransmit to fulfill the slot. If the packets need-to-retransmit can not fulfill the page, the sender uses packets of the next task.

We set different packet loss rates and evaluate the performance of different transferring schemes. We record the mean number of transmissions for the two tasks with different schemes and the total number of transmissions needed at different packet loss rates. The fewer transmissions consumed during the bulk data transfer, the higher energy efficiency it achieves.

4.2 Simulation Results

As shown in Figure 3, we can see the mean transfer rounds of offloading tasks by using different transfer schemes at different loss packet rates. From the figure we can see that, with the increasing packet loss rate, the mean transfer rounds increase greatly for all schemes. Specifically, in Figure 3(a), we can see that there is not an apparent gap between our transfer scheme and ULTRA in transferring the total two tasks, but both of them outperform the strictly sequential scheme, especially when the packet loss rate reaches a high level. The main reason is that both our transfer scheme and ULTRA fulfill every transmission slot, so that the utilization of every page can be improved and the total transmissions can be reduced. But when we turn our eyes to the task 1 which is with a high priority, we can see our transfer scheme has a better performance than the other two schemes in Figure 3(b) as task 1 finishes earlier in the proposed work. This is because our transfer scheme tends to priorly retransfer the lost packets from the high priority for ensuring that the high priority lost packets can be received successfully as soon as possible. Thus we can ensure that the high priority tasks can be offloaded as soon as possible. The strict priority sacrifices some transmission rounds of task 2 as its packets are postponed to the later slots. In Figure 3(c), we can see that our scheme has slightly more transfer rounds than ULTRA in offloading task 2, but still far smaller than the strictly sequential scheme.

Considering IoT devices also are energy-constrained, we evaluate the energy consumption in transferring by accounting the mean packet number needed to be transferred. As we can see in Figure 4, with increasing packet loss rate,

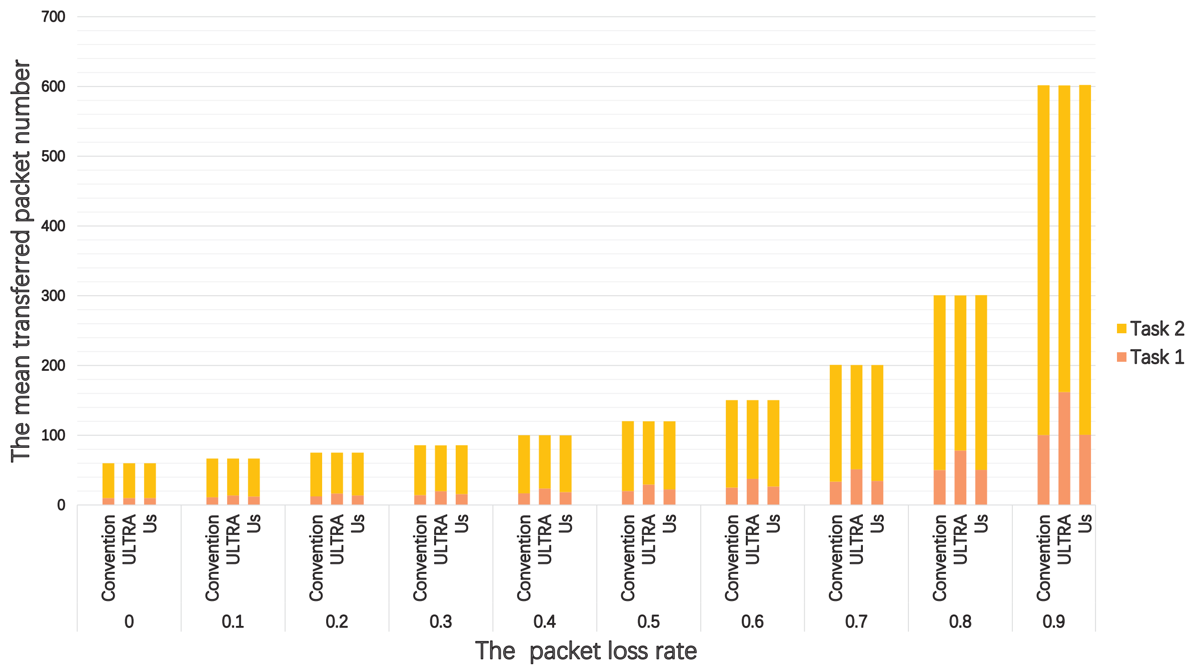


Figure 4. The mean transfer packets number in different transfer scheme

the total number of transmissions for all schemes increase greatly. However, all the schemes transfer the two tasks with similar transmission counts when packet loss rate is constant. For ULTRA and strictly sequential scheme, the reason is that both of them retransfer a packet once in one transfer slot until the packet is successfully delivered. Thus the sum of packets needed to be transferred is approximately equal for the two transfer schemes. For our transfer scheme, the reason is that we determine the number of retransmissions for the same lost packet in one transmission slot by calculating the expected number of transmissions required by the link. As a result, our transfer scheme is more likely to successfully deliver the lost packets in fewer transfer slots, and thus there is not extra energy consumption compared to ULTRA and conventional transfer schemes.

The simulation results show that our transfer scheme achieves a better tradeoff between task priorities and transmission efficiency without introducing extra energy consumption, while ULTRA fails to guarantee the priorities and the strictly sequential scheme wastes too many transmission rounds.

Discussion. Considering the ongoing trend on the mobile edge computing, where the wireless nodes typically have direct links to the edge servers, the need for multi-hop transmissions may not exist in many scenarios. As a result, the three-slot paradigm may be changed in these scenarios. Our future work will focus on adapting our scheme to the single-hop but high-density networks.

5 Conclusion

In this paper, we investigate the problem of bulk data transferring for task offloading. We propose an novel scheme that considers both priorities and transmission efficiency. According to the task priorities and the time-varying packet loss rate, the proposed scheme can organize transmissions in each round with priority to high-priority task packets and without strict packet-level orders. We conduct simulation experiments to evaluate the performance of the proposed transfer scheme and compare it to the existing works, UlTRA and strictly sequential scheme. Simulation results show that our proposed transfer scheme can achieve a good tradeoff between task priorities and transmission efficiency, without incurring extra energy consumption. The problem is that it might cause more transfer delay for low priority tasks compared to ULTRA. Our future work will focus on adapting our scheme to the single-hop but high-density networks, which is the more typical scenario in wireless edge computing.

6 Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61602095), the Fundamental Research Funds for the Central Universities (No. ZYGX2016KYQD098 and No. ZYGX2016J079), National Postdoctoral Program for Innovative Talents of China (No. BX201700046), and the EU FP7 CLIMBER project under Grant Agreement No. PIRSES-GA-2012-318939.

7 References

[1] M. Aazam, S. Zeadally, and K. A. Harras. Offloading in fog computing for iot: Review, enabling technologies, and research opportunities. *Future Generation Computer System*, 2018.

[2] Y. Cao, N. Wang, Z. Sun, and H. Cruickshank. A reliable and efficient encounter-based routing framework for delay/disruption tolerant networks. *IEEE Sensors Journal*, 15(7):4004–4018, 2015.

[3] W. Dong, Y. Liu, C. Wang, X. Liu, C. Chen, and J. Bu. Link quality aware code dissemination in wireless sensor networks. In

IEEE International Conference on Network Protocols, pages 89–98, 2011.

[4] Gubbi, Jayavardhana, Marusic, Slaven, Palaniswami, Marimuthu, Buyya, and Rajkumar. Internet of things (iot): A vision, architectural elements, and future;directions. *Future Generation Computer Systems*, 29(7):1645–1660, 2013.

[5] A. Hagedorn, D. Starobinski, and A. Trachtenberg. Rateless deluge:over-the-air programming of wireless sensor networks using random linear codes. In *International Conference on Information Processing in Sensor Networks*, pages 457–466, 2008.

[6] I. H. Hou, Y. E. Tsai, T. F. Abdelzaher, and I. Gupta. Adapcode: Adaptive network coding for code updates in wireless sensor networks. In *INFOCOM 2008. the Conference on Computer Communications. IEEE*, pages 1517–1525, 2007.

[7] L. Huang and S. Setia. Cord: Energy-efficient reliable bulk data dissemination in sensor networks. *Proceedings - IEEE INFOCOM*, 14(3):115–124, 2008.

[8] J. W. Hui and D. Culler. The dynamic behavior of a data dissemination protocol for network programming at scale. In *The ACM Conference on Embedded Networked Sensor Systems*, pages 81–94, 2004.

[9] S. Kulkarni and L. Wang. *Energy-efficient multihop reprogramming for sensor networks*. ACM, 2009.

[10] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, and E. Brewer. *TinyOS: An Operating System for Sensor Networks*. Springer Berlin Heidelberg, 2005.

[11] P. Levis, N. Patel, D. Culler, and S. Shenker. Trickle: a self-regulating algorithm for code propagation and maintenance in wireless sensor networks. In *Conference on Symposium on Networked Systems Design and Implementation*, pages 2–2, 2004.

[12] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal*, 4(5):1125–1142, 2017.

[13] Y. Liu, G. Zhou, J. Zhao, G. Dai, X. Y. Li, M. Gu, H. Ma, L. Mo, Y. He, and J. Wang. Long-term large-scale sensing in the forest: recent advances and future directions of greenorbs. *Frontiers of Computer Science in China*, 4(3):334–338, 2010.

[14] B. Martinez, M. Monton, I. Vilajosana, and J. D. Prades. The power of models: Modeling power consumption for iot devices. *IEEE Sensors Journal*, 15(10):5777–5789, 2015.

[15] D. Min, Z. Xiao, B. Sheng, Q. Huang, and X. Pan. Design and implementation of heterogeneous iot gateway based on dynamic priority scheduling algorithm. *Transactions of the Institute of Measurement & Control*, 36(7):924–931, 2014.

[16] D. Moss, J. Hui, P. Levis, Choi, and J. Il. Cc2420 radio stack.

[17] V. Naik, A. Arora, P. Sinha, and H. Zhang. Sprinkler: a reliable and energy efficient data dissemination service for wireless embedded devices. In *IEEE International Real-Time Systems Symposium*, pages 277–286, 2005.

[18] V. Naik, A. Arora, P. Sinha, and H. Zhang. Sprinkler: A reliable and energy efficient data dissemination service for extreme scale wireless networks of embedded devices. *IEEE Transactions on Mobile Computing*, 6(7):777–789, 2007.

[19] G. Premsankar, M. D. Francesco, and T. Taleb. Edge computing for the internet of things. *IEEE Internet of Things Journal*, 5(2):1275–1284, 2018.

[20] M. Rossi, N. Bui, G. Zanca, L. Stabellini, R. Crepaldi, and M. Zorzi. Synapse++: Code dissemination in wireless sensor networks using fountain codes. *IEEE Transactions on Mobile Computing*, 9(12):1749–1765, 2010.

[21] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, 2016.

[22] S. Wang, A. Basalamah, S. M. Kim, S. Guo, Y. Tobe, and T. He. Link-correlation-aware opportunistic routing in wireless networks. *IEEE Trans. Wireless Communications*, 14(1):47–56, 2015.

[23] Z. Zhao, J. Bu, W. Dong, T. Gu, and X. Xu. Coco+: Exploiting correlated core for energy efficient dissemination in wireless sensor networks. *Ad Hoc Networks*, 37:404–417, 2016.

[24] Z. Zhao, W. Dong, J. Bu, and T. Gu. Exploiting link correlation for core-based dissemination in wireless sensor networks. In *Eleventh IEEE International Conference on Sensing, Communication, and Networking*, pages 372–380, 2014.

[25] Z. Zhao, W. Dong, J. Bu, T. Gu, and G. Min. Accurate and generic sender selection for bulk data dissemination in low-power wireless networks. *IEEE/ACM Transactions on Networking (ToN)*, 25(2):948–959, 2017.

[26] Z. Zhao, W. Dong, J. Bu, Y. Gu, and C. Chen. Link-correlation-aware data dissemination in wireless sensor networks. *IEEE Transactions on Industrial Electronics*, 62(9):5747–5757, 2015.

[27] Z. Zhao, W. Dong, G. Chen, G. Min, T. Gu, and J. Bu. Embracing corruption burstiness: Fast error recovery for zigbee under wi-fi interference. *IEEE Transactions on Mobile Computing*, 16(9):2518–2530, 2017.

[28] Z. Zhao, W. Dong, G. Guan, and J. Bu. Modeling link correlation in low-power wireless networks. In *IEEE Conference on Computer Communications*, pages 990–998, 2015.

[29] Z. Zhao, W. Dong, G. Min, G. Chen, T. Gu, and J. Bu. Towards repeatable wireless network simulation using performance aware markov model. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 2168–2176. IEEE, 2018.

[30] Z. Zhao, G. Min, W. Gao, Y. Wu, H. Duan, and Q. Ni. Deploying edge computing nodes for large-scale iot: A diversity aware approach. *IEEE Internet of Things Journal*, 2018.

[31] Z. Zhao, Z. Wang, G. Min, and Y. Cao. Highly-efficient bulk data transfer for structured dissemination in wireless embedded network systems. *Journal of Systems Architecture*, 72:19 – 28, 2017. Design Automation for Embedded Ubiquitous Computing Systems.

[32] X. Zheng, Z. Cao, J. Wang, Y. He, and Y. Liu. Zisense: Towards interference resilient duty cycling in wireless sensor networks. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*, pages 119–133. ACM, 2014.

[33] X. Zheng, Z. Cao, J. Wang, Y. He, and Y. Liu. Interference resilient duty cycling for sensor networks under co-existing environments. *IEEE Transactions on Communications*, 65(7):2971–2984, 2017.

[34] X. Zheng, J. Wang, W. Dong, Y. He, and Y. Liu. Bulk data dissemination in wireless sensor networks: analysis, implications and improvement. *IEEE Transactions on Computers*, 65(5):1428–1439, 2016.

[35] T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang. Achieving efficient flooding by utilizing link correlation in wireless sensor networks. *IEEE/ACM Transactions on Networking (TON)*, 21(1):121–134, 2013.