Competition: Wireless-Transparent Sensing Platform

Chun-Hao Liao, Theerat Sakdejayont, Makoto Suzuki, Yoshiaki Narusue, and Hiroyuki Morikawa
School of Engineering, The University of Tokyo
7–3–1, Hongo, Bunkyo-ku, Tokyo 113–8656, Japan
{liao, theerat, makoto, narusue, mori}@mlab.t.u-tokyo.ac.jp

Abstract

In this abstract, we introduce our published work, Wireless-Transparent Sensing Platform (WTSP) and the improvements we made toward the EWSN dependability competition 2018. WTSP is a TDMA-based protocol adopting the concurrent transmission flooding (CTF). WTSP has two unique features on time slot scheduling. First, the sink node schedules slots on the fly; only several slots are determined within a schedule packet, and additional schedule packets are distributed once the distributed schedule is completed, making scheduling very flexible. Second, the scheduling is service-driven; scheduling is delegated to upper-layer services and each service directly makes a schedule, which allows the sink node to adapt to communication demands accurately. Given the multi-source multi-destination traffic pattern and harsh RF environments in the EWSN competition, several refinements have been made on WTPS including *1)* a dedicated service that realizes UDP-like one-way streaming dissemination, *2)* special forwarding nodes to allow progressive forwarding in CTF, and *3)* adding source redundancy to improve the utility, reliability, as well as the latency performance.

1 Introduction

In this abstract, we describe Wireless-Transparent Sensing Platform (WTSP), a data collection platform based on the concurrent transmission flooding. Our experience in many cooperative research projects tells us that it is usually the lossy, unstable, and opaque nature of wireless networks that hinders wireless sensor networks (WSN) from being adopted in real applications. Given this, we have strived to develop a platform that can make the wireless communication *transparent* to the users. The WTSP achieves this goal by realizing end-to-end retransmission, data synchronization, responsiveness, and adaptability to various traffic demands. The details of WTSP can be found in [1].

WTSP adopts the concurrent transmission flooding (CTF) [2], which is a very efficient flooding scheme that has many merits: it can not only spread a packet to the whole network very quickly but also achieve time synchronization easily. These merits help to reduce the protocol complexity considerably. Moreover, WTSP introduces an *on-the-fly* (OTF) and *service-driven* scheduler on CTF-based networks.

OTF scheduling: The sink node performs OTF scheduling to ensure the flexibility. In WTSP, time is divided into slots with a length of some ten milliseconds. The sink node communicates with non-sink nodes in a request-response manner continuously. The sink node requests with a control packet, which specifies the schedule of the following several slots, and non-sink nodes reply in accordance with the control packet. Right after the current schedule is completed, the sink node distributes a new control packet if further communications are necessary, or distributes sleep packets otherwise.

Service-driven scheduling: Moreover, in order to adapt to the communication demands accurately, we designed the scheduling as *service-driven*. WTSP is designed in a layered fashion and consists of a super scheduler called `ctfnetd` and several network services, including versatile collection, stream-oriented dissemination to issue user commands, bulk dissemination for reprogramming, and ping for a quick reachability check. `ctfnetd` arbitrates network services, and network services have their own slot scheduler. This isolation makes it possible to design each of the schedulers independently.

These two scheduling techniques offer great benefits. For example, the sink node assigns slots in a row for maximizing collection throughput when the traffic demand is high and instructs non-sink nodes to sleep for ensuring energy efficiency when there are no traffic demands. Also, if a packet is lost, the sink node can request a retransmission at the next schedule.

2 Toward EWSN Competition

Thanks to organizers, in this year, the competitors are allowed to access the testbed remotely and given a long preparation period for information gathering and protocol improvements. In this section, we would like to first discuss the challenges that WTSP faces in this competition and how we improve WTSP to respond to these challenges.

International Conference on Embedded Wireless Systems and Networks (EWSN) 2018
14–16 February, Madrid, Spain

ISBN: 978-0-9949886-2-1

2.1 Challenges of EWSN Testbed

2.1.1 Multi-source multi-destination scenario

WTSP is originally designed for realizing two main traffic patterns: 1) data collection mode that collects the sensor data from multiple non-sink nodes to a sink node, and 2) data dissemination mode that distributes the command data from the sink node to multiple non-sink ones. On the other hand, this year's competition introduces a complicated scenario where the messages from multiple source nodes need to be passed to multiple specified destination nodes.

2.1.2 Harsh RF environments

WTPS adopts CTF as a base. Despite all its merits, CTF suffers from a limitation which we refer to as *volatile relaying*. To elaborate, common CTF-based protocols slot individual packet transmission in a TDMA fashion. In each slot, a source node kicks off a transmission as a broadcast, and any nodes hearing the packet would relay that as another broadcast right after the reception to allow the packet flooding through the network. Due to this broadcasting nature, there is no ACK between each relay, and when the slot ends, non-destination nodes would simply flash out its radio. Consequently, if a packet fails to reach its destination in a slot, the source node needs to start all over again in a later slot as if all the relaying efforts in this slot are just evaporated.

In a network with decent link quality, the loss of such limitation is minor, and the proposed end-to-end retransmission scheme in [1] can ensure the reliability with only negligible overheads. However, the scenarios in the EWSN competition are very harsh. Our experiments results indicate that, in higher-level jamming cases, hardly can any flooding successfully penetrate the network and reach the destination. In most of the cases, only local packet exchanges can occasionally succeed.

For a protocol adopting CTF, such interference is critical. This is not only because most of the attempts to retransmit data packet ends in vain due to the volatile relaying. What more lethal is that even the control packet cannot properly be delivered to end nodes, which results in a failure of synchronizations and hence TDMA scheduling.

2.2 Protocol Improvement for EWSN

To overcome the aforementioned difficulties, several improvements have been implemented on WTPS, which we explain as follows.

2.2.1 UDP-like one-way streaming service

Toward EWSN competition, we implement a simple but efficient service to achieve the multi-source to multi-destination transmission. As we have mentioned, in harsh RF environment, the reliability can hardly be guaranteed even with end-to-end re-transmission. Therefore, the new service realizes a one-way data dissemination without end-to-end ACK. Specifically, the slots are assigned to each source node (and also special forwarding nodes that will be mentioned later) in a simple round-robin manner, and the destination nodes keep updating the GPIO as long as a packet is being received. The design concept is very similar to the online streaming using the UDP protocol.

2.2.2 Progressive forwarding

In order to overcome the volatile relaying limitation of CTF, we strive to make the packets being able to *progressively* move forward over the slots. To achieve this, we introduce so-called *forwarding nodes* to the protocol. These nodes are programmed to have the capability of buffering the received packets, and slots are also assigned to them for forwarding the buffered packets. Moreover, the packet transmitted by these forwarding nodes would be also used for synchronization.

Adding such forwarding nodes also introduces a new inconsistency issue. It happens when a destination node first receive a fresh packet directly from a source node, while later receives an outdated packet from a forwarding node. To tackle such inconsistency, we embedded the timestamp information of GPIO into the packet, and the node will only update the buffer or GPIO output when newer information is received.

Moreover, it should be also noted that the number of the forwarding nodes are of great importance to the reliability and latency performance. Since the forwarding nodes occupy slots, their number needs to be constrained so that the latency performance is not significantly affected.

2.2.3 Source redundancy

In our UDP-like streaming service implementation, the effective data per transmission is actually only 1 bytes (1 bits for GPIO, and 7 bits for the timestamp information). Comparing with this, the overhead of IEEE 802.15.4 packets, such as preamble, header, and CRC, are relatively large. In order to increase the utility while improving the reliability and latency performance, we allow a packet to carry the information from multiple sources so that one successful packet reception yields update of multiple data streams. Specifically, not only aforementioned forwarding nodes, all slot-owning nodes including the sources are assigned to buffer and transmit the GPIO information for multiple specified nodes.

2.2.4 Other techniques

In addition to the aforementioned techniques, we also adopt other techniques for anti-jamming, such as channel hopping.

Acknowledgments

This work was supported by Cross-ministerial Strategic Innovation Promotion Program "Infrastructure Maintenance, Renewal and Management Technology" and JSPS Grant-in-Aid for Scientific Research (A) Grant Number JP16H02358.

3 References

[1] Makoto Suzuki, Chun-Hao Liao, Sotaro Ohara, Kyoichi Jinno, and Hiroyuki Morikawa. Wireless-transparent sensing. In *Proc. International Conference on Embedded Wireless Systems and Networks (EWSN)*, pages 66–77, Feb. 2017.

[2] Federico Ferrari, Marco Zimmerling, Lothar Thiele, and Olga Saukh. Efficient network flooding and time synchronization with Glossy. In *Proc. 10th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*, pages 73 –84, Apr. 2011.