



An Error Recovery Mechanism for Wireless Sensor Networks

Dong-Il Kim*, Member, KIICE

Department of Information and Communication, Dong-Eui University, Busan 614-714, Korea

Abstract

In wireless sensor networks, the importance of transporting data correctly with reliability is increasing gradually along with the need to support communications between the nodes and sink. Data flow from the sink to the nodes requires reliability for control or management that is very sensitive and intolerant of error; however, data flow from the nodes to the sink is relatively tolerant. In this paper, with emphasis on the data flow from the sink to the nodes, we propose a mechanism that considers accurate transport with reliability hop-by-hop. During the process of sending the data, if errors occur or data is missing, the proposed mechanism supports error recovery using a fixed window with selective acknowledgment. In addition, this mechanism supports congestion control depending on the buffer condition. Through the simulation, we show that this mechanism is accurate, reliable, and proper for transport in wireless sensor networks.

Index Terms: Error recovery mechanisms, Wireless sensor networks

I. INTRODUCTION

Wireless sensor networks (WSNs) have met huge growth and have significant future prospects of evolution with a wide range of potential applications including environmental monitoring, military applications, medical systems, smart vehicles, and robotic exploration. Therefore, WSNs will play an important role in making the goal of ubiquitous computing a reality. The differences between traditional network products or technologies like Internet protocol (IP) networks makes research related to WSNs more attractive.

Unlike traditional IP networks, accurate and reliable transport is still an open research question in the context of WSNs and there has been little work on the design of such transport for WSN.

This is not surprising because the vast majority of WSN applications do not require accurate and reliable transport. Currently, WSNs tend to be application-specific and are typically hard-wired to perform a specific task efficiently at low cost. Because of the application-specific environment of WSNs, it is difficult to design transport that can be optimized for every application.

We propose a mechanism for accurate transport with congestion control that is not application-specific and satisfies the general constraints in WSNs, which are a limited source of power, unreliable radio link, memory constraints, and limited computational ability.

Communications from the nodes to the sink are already fairly reliable, so we focus on the data flow from the sink to the nodes. Due to control and management activities such as re-tasking and re-programming, data

Received 11 July 2012, Revised 24 July 2012, Accepted 31 July 2012

* Corresponding Author E-mail: dikim@deu.ac.kr

Open Access <http://dx.doi.org/10.6109/jicce.2012.10.3.237>

print ISSN:2234-8255 online ISSN:2234-8883

© This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Copyright © The Korea Institute of Information and Communication Engineering

flow from the sink to the nodes is both critical and error intolerant. On the other hand, data flow from the nodes to the sink is relatively tolerant to error.

This paper is organized as follows. Section II classifies and compares the existing protocols that have a mechanism for accurate and reliable transport for WSNs. Section III describes the proposed mechanism. Section IV presents the simulation results. Finally, we have some concluding remarks and a description of future work in Section V.

II. RELATED WORK

In this section, we classify and compare the existing protocols that provide accurate and reliable transport for WSNs. They aim at a reliability guarantee and congestion control upstream (from the nodes to the sink) or downstream (from the sink to the nodes)

A. Pump-Slowly, Fetch-Quickly (PSFQ)

PSFQ aims to distribute data from the sink to the nodes by pacing data at a relatively slow-speed ("pump slowly") but allowing nodes that experience data loss to fetch (recover) any missing segments from immediate neighbors very aggressively (local recovery, "fetch quickly") [1].

The protocol aims at a downstream reliability guarantee. The motivation of PSFQ is to achieve loose delay bounds while minimizing the loss recovery cost by localized recovery of data among immediate neighbors. It contains three components: pump operation, fetch operation, and report operation. Firstly, the sink slowly broadcasts a packet (with such fields as file ID, file length, sequence number, time-to-live [TTL], and report bit) to its neighbors every time T until all the data fragments have been sent out. Secondly, a node can go into fetch mode once a sequence number gap in a file fragment is detected and issue negative acknowledgment (NACK) in a reverse path to recover the missing fragment. The NACK does not need to be relayed unless the number of times the same NACK is heard exceeds a predefined threshold while the missing segments requested by the NACK message are no longer retained in a node's cache. Thirdly, the sink can make nodes provide feedback data delivery status information to the sink through a simple and scalable hop-by-hop report mechanism.

B. Reliable Multi Segment Transport (RMST)

RMST belongs to an upstream reliability guarantee. It is designed to run above directed diffusion [2] (to use its discovered path from the nodes to the sink) in order to provide guaranteed reliability from the nodes to the sink (delivery and fragmentation/reassembly) for applications [3]. RMST is a selective NACK-based protocol. RMST basically operates as follows. Firstly, RMST uses a timer-driver mechanism to detect data loss and send NACK on the way from the detecting node to the sources (cache or non-cache mode). Secondly, NACK receivers are responsible for looking for the missing packet, or forwarding NACK on the path toward the sink if it fails to find the missing packet or is in non-cache mode.

C. Event-to-Sink Reliable Transport (ESRT) in WSN

ESRT aims at providing reliability from the nodes to the sink and congestion control simultaneously. It aims for an upstream reliability guarantee [4]. Firstly, it needs to periodically compute the factual reliability r according to the number of the successfully received packets in a time interval. Secondly, and most importantly, ESRT deduces the required sensor report frequency f from r : $f = G(r)$. Thirdly and finally, ESRT informs all nodes of f through an assumed channel with high power and nodes can even report and transmit packets with frequency f . ESRT is an end-to-end approach to guaranteeing a desired reliability by regulating sensor report frequency. It provides reliability for applications, not for each single packet. The additional benefit resulting from ESRT is energy-conservation since it can control sensor report frequency.

Table 1. Comparison of existing transport protocols

Attributes	PSFQ	RMST	ESRT
Reliability	Yes	Yes	Yes
Congestion control	No	No	Passive
Congestion detection	No	No	Buffer size
Flow direction	Sink to nodes	Nodes to sink	Nodes to sink
EtE/HbH	HbH	HbH	EtE
ACK/NACK	NACK	NACK	ACK
Sequence number	Yes	Yes	No

PSFQ: pump-slowly, fetch-quickly, RMST: reliable multi segment transport, ESRT: event-to-sink reliable transport, ACK: acknowledgment, NACK: negative ACK.

The classification and comparison show that the major functions of accurate and reliable transport for WSNs are reliability guarantee and congestion control.

III. PROPOSED MECHANISM DESCRIPTION

The proposed mechanism pursues a reliability guarantee and congestion control simultaneously. To meet these conditions, this mechanism has the operations of handshaking, selective acknowledgement hop-by-hop, and congestion control by checking the buffer status and timer.

A. Handshaking

There are several routing protocols for WSNs. Among them, our mechanism uses the dynamic source routing (DSR) protocol that is an on-demand routing algorithm [5].

DSR is a sink-initiated algorithm that is appropriate for infrequent and one-time use. First, a sink node broadcasts the “handshaking message” (Hsk_Msg) to others.

A recipient of the Hsk_Msg keeps broadcasting to other nodes. To avoid repeated broadcasting, a sequence number is required. When the Hsk_Msg reaches the destination node, the latter sends back the “handshaking message to respond” (Hsk_Rsp) to the sink node using the route that is created during the first half of the process. By the mechanism of broadcasting, the packet with the shortest path will arrive at the destination first. Therefore, the shortest path from the source to the destination is guaranteed. After transmitting data, the sink node sends an “end message” (End_Msg) that signifies that the sink has no data anymore to send to the destination node. The destination nodes send back an “end message to respond” (End_Rsp) to the sink node.

B. Selective Ack Hop -by-hop

To reduce the redundancy and recover error are also important factors of our mechanism. There are two approaches to reduce redundancy and increase error recovery. First, the hop-by-hop manner, which is proposed in the paper on PSFQ, is appropriate in a high error environment. The error rate in WSNs is geometrically cumulative. The hop-by-hop manner that shows a linear increment of the error rate mitigates this situation in high error environments. Any data or control packets are checked for their validity when they go through each hop. Second is the selective repeat (SR) mechanism [6]. SR was developed for the traditional internet to increase the throughput and decrease redundancy. By a modified SR mechanism, our mechanism decreases the number of control (ACK or NACK) packets. Specifically, this uses selective acknowledgements using a fixed window. Control packets include the sequence base (the largest sequence number without any missing packets) and acknowledged sequence numbers. By this mechanism, the number of control packets is decreased and the mechanism actually works as NACK requests.

For example, in Fig. 2, the sink sends data with a sequence number (SN) to the destination node hop-by-hop. Between the second hop node and the destination node, the SN₃ data packet is lost. However, the destination node receives data until the SN₄ packet for sending the ACK message to the second hop node. The ACK message consists of a fixed window that presents a pair of SNs that denotes that the destination node received Data SN₂ and SN₄, but not SN₃. The second node then receives the ACK 2, 4 and retransmits Data SN₃.

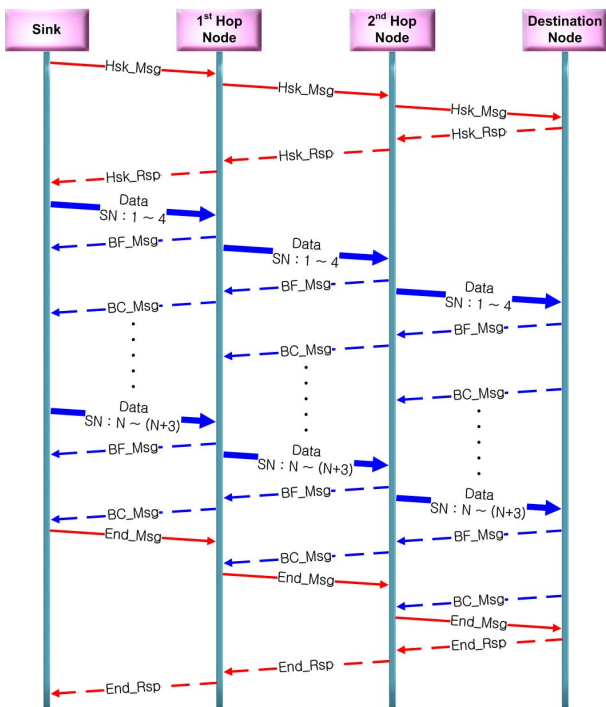


Fig. 1. Data transmission procedure in mechanism. Hsk_Msg: handshaking message, Hsk_Rsp: handshaking message to respond, End_Msg: end message, End_Rsp: end message to respond, BC_Msg: buffer cleared message, BF_Msg: buffer full message, SN: sequence number.

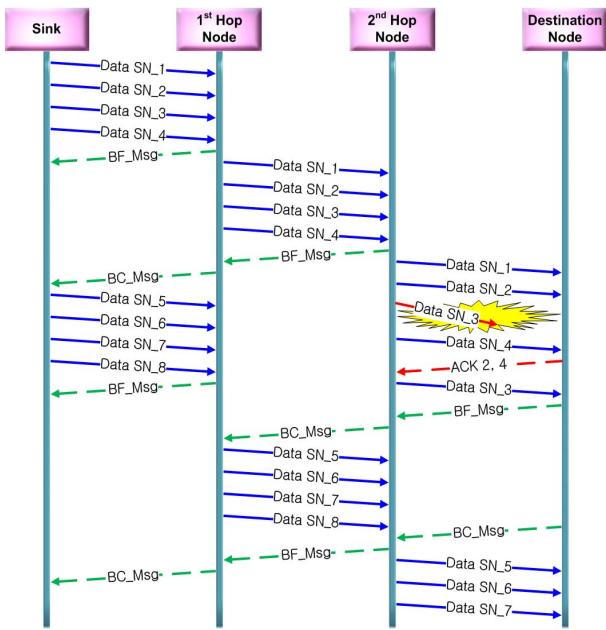


Fig. 2. Recovery procedure in mechanism.

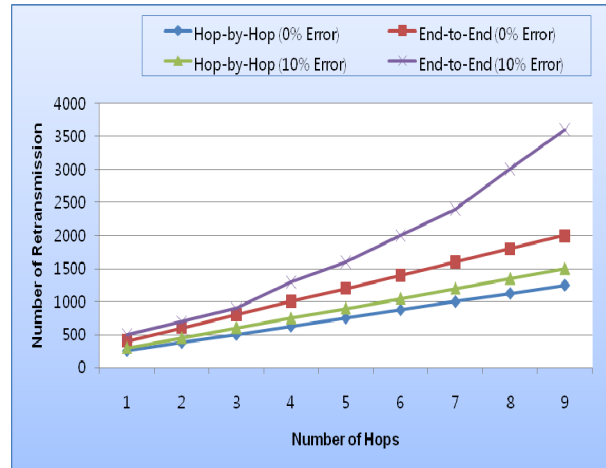
C. Congestion Control

Congestion control is provided by two mechanisms: checking the buffer status and the timer. Intermediate nodes have a relatively small buffer. If a buffer in an intermediate node is full, then the node will send a “buffer full message” (BF_Msg) to its predecessor (parent node). If the node has buffer space enough to receive additional messages, it will then send a “buffer cleared message” (BC_Msg). In a highly error-prone environment, these types of control messages can be missing. This disturbs further transmission. To avoid missing this control packet, the nodes will check that these control messages are delivered correctly. If there is no response after a time predetermined by the timer, then it re-sends the control messages. If packet is missing due to congestion, the nodes send an ACK message. After that, the predecessor node re-sends the missing packets that were lost by congestion.

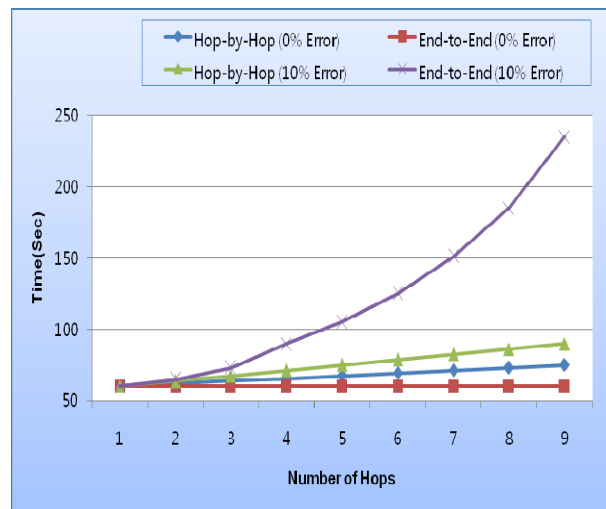
IV. SIMULATION

In the simulation, we used TOSSIM [7] in TinyOS [8] and focused on a comparison between end-to-end and hop-by-hop mechanisms. In Fig. 3 shows a no error and a 10% error environment were used for sending 100 packets for 9 hops. We measured the number of retransmissions and the throughput at each hop. Though the throughput of the end-to-end mechanism is better

without errors, as a whole, performance of the hop-by-hop mechanism is shown to be better in error prone environments. This would be expected in practical WSNs.



(a)



(b)

Fig. 3. Number of retransmissions in number and 10% error (end-to-end vs. hop-by-hop).

V. CONCLUSIONS

Our proposed mechanism using a selective ACK hop-by-hop and congestion control method is suitable for WSNs in satisfying a reliability guarantee and congestion control. This mechanism is designed for WSNs in which the intermediate nodes do not require having a solution for redundancy of all of the data. In a future work, we will investigate applying security to the transport layer, and was also have studies in progress using another

media access control protocol and proper routing protocol. Finally, we plan to study a detailed error recovery mechanism and protocol for WSNs.

ACKNOWLEDGMENTS

This work was supported by a Dong-Eui University Grant (2011AA170).

REFERENCES

- [1] C. Y. Wan, A. T. Campbell, and L. Krishnamurthy, "PSFQ: a reliable transport protocol for wireless sensor networks," *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, Atlanta, GA, pp. 1-11, 2002.
- [2] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, Boston, MA, pp. 56-67, 2000.
- [3] F. Stann and J. Heidemann, "RMST: reliable data transport in sensor networks," *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, Anchorage, AK, pp. 102-112, 2003.
- [4] Y. Sankarasubramaniam, O. B. Akan, and I. F. Akyildiz, "ESRT: event-to-sink reliable transport in wireless sensor networks," *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Annapolis, MD, pp. 177-188, 2003.
- [5] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, Boston, MA: Kluwer Academic Publishers, 1996.
- [6] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP selective acknowledgment options," The Internet Engineering Task Force, Fremont: CA, *RFC 2018*, 1996.
- [7] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: accurate and scalable simulation of entire TinyOS applications," *Proceedings of the 1st International Conference on Embedded Networked Sensor System*, Los Angeles, CA, pp. 126-137, 2003.
- [8] TinyOS [Internet]. Available: <http://www.tinyos.net/>.



Dong-II Kim

received the B.S., M.S., and Ph.D. degrees from Kwangwoon University, Korea. Since 1991, he has been at Dong-Eui University, Korea, where he is a professor in the Information and Communication Engineering Department. From 1983 to 1991, he was a general manager at the switching research center in LG Information and Communication Institute. He was a visiting researcher at the Standardization Research Center of ETRI from 1998 to 1999. He has been the chief of the Computer Center at Dong-Eui University and an international expert in the field of ICT at the Korea Communications Commission (KCC). His research interests are analysis of performance in communication networks, protocols in wireless networks, and standardization of ICT.