A Semi-Distributed Reliable Data Transport in Sensor Networks

Md. Obaidur Rahman, Muhammad Mostafa Monowar, Choong Seon Hong
Kyung Hee University.

rupam@networking.khu.ac.kr, monowar@networking.khu.ac.kr, cshong@khu.ac.kr

Abstract

This paper presents a semi-distributed solution for reliable data delivery in sensor networks. Unlike end-to-end or hop-by-hop protocols, the proposed work incorporates network partitioning and aims for contention-free and delay minimized retransmission for the lost data packets. The simulation result establishes the effectiveness of the proposed work in terms of e-2-e delay.

Ⅰ. Introduction

Many sensor network applications (i.e., surveillance detection) demand in-order and 100% delivery of the generated packets to detect any event reliably and successfully. Currently in literature there exist mainly two types of reliability control protocol for wireless sensor networks (WSNs): (i) End-to-end centralized approach, and (ii) Hop-by-hop protocol distributed solution.

Despite the advantageous properties (i.e., energy saving) of NACK-based end-to-end retransmission, existing reliable protocols like RCRT [3] cannot capitalize its full benefit. Because in RCRT each retransmission is treated as a normal transmission, hence packet loss probability still exists for the retransmitted packet due to contention and collision. On the other hand, a distributed ACK based solution PSFQ [2] encounters ACK explosion problem, increases delay, and cannot ensure in-order delivery of the packets. However, we argue that a reliable protocol should cope-up all these problems and ensure delay minimized and contention-free retransmission of the lost data. Furthermore, a reservation based schedule [1] only can guarantee prioritized transmission of lost data.

This paper presents a reliable data transport protocol which can overcome the above mentioned problems by partitioning the network and making reservation based retransmission schedule. This work can achieve maximum channel capacity in each network partition in accessing the medium and ensure collision free retransmission. Moreover, it significantly reduces the delay.

"This research was supported by the MKE (Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement)" (NIPA-2010-(C1090-1021-0003)) Dr. CS Hong is the corresponding author.

Ⅱ. Proposed Scheme

A. Network Model

We assume a WSN having a number of sensor nodes, with a sink (i.e., receiver) and multiple source nodes. The network scenario is many-to-one that converges to a tree-topology rooted at the sink, and all the other nodes places on different levels of the tree.

In the subsequent sections we use the notations: let, each sensor node is denoted by n, and the source and sink (receiver) are expressed as S and R respectively. A pre-established path $P_{(S,R)}$ is assumed between each source and sink, and intermediate nodes on $P_{(S,R)}$ have a level value L_i, where starting from the sink at first level $L_0 = 0$, at second level $L_1 = 1$ and at i-th level $L_i = l$.

B. Semi-Distributed Network Partitioning

At the beginning we partition the network as shown in Fig. 1. The network partitioning is used here to avoid the problems of fully distributed and end-to-end approaches [2][3][4]. Furthermore, the effect of interference is also taken into account to achieve better channel utilization in each partition of the network, so that nodes on different partitions can transmit simultaneously. More specifically in order to avoid the intra-flow contention in retransmission this partitioning method ensures maximum throughput.

In order to partition the network we use the equation 1 for each node, where I is the interference range, and here I=2.

$$Part_i = L_i \bmod (I+1) \qquad i = 0, 1, 2, \ldots, l \qquad (1)$$

Here, $Part_i$ is a value of the nodes of different level and used in network partitioning. As shown in Fig. 1, each partition has an upstream partition node P_u, a downstream

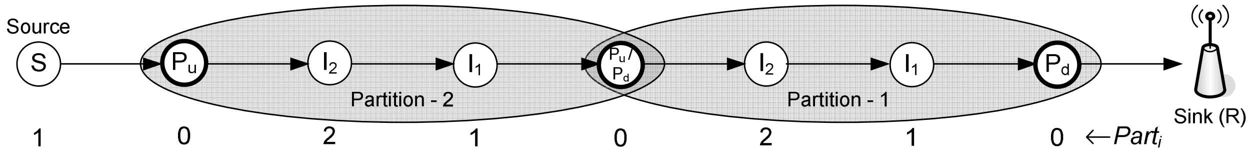

Fig. 1: Network partitioning of linear path $P_{(S,R)}$. The corresponding $Part_i$ values of each node are shown under every node.

partition node P_d and intermediate nodes I_1 and I_2. A common node at the intersection of two partitions acts as both P_u and P_d.

C. Reliability Control with MAC Sequencing

In the proposed protocol, each flow f_S is forwarded through node P_u, and each transmitted packet are stored in the retransmit buffer of P_u.

In contrast, a P_d node on the same partition receives all the packets and checks the packet sequence id $ID_{f(s)}$, assigned by P_u. However, whenever a gap is detected in the packet sequence, P_d issues a NACK and sends the lost packet list as well as retransmission request to P_u.

Furthermore, P_u stores the packets until it receives a B-ACK (Block ACK) from P_d having the highest sequence id of the in-order received packets. A node P_d issues a B-ACK whenever the number of in-order received packets crosses half ($1/2$) of the retransmit-buffer. However, upon receiving a NACK with retransmission request, P_u announces a channel reservation schedule in its next data transmission. Nodes I_2, I_1 and P_d further broadcast the reservation schedule for the same packet so that it can either receive or receive-transmit the retransmitted data avoiding contention. The pseudo code in Algorithm 1 includes the operation of the reliability control protocol.

Algorithm 1: Semi-Distributed Reliability Control

```
for each node n
    if {n = Pu}
        Assign ID_f(S) at each packet p(ID_f(S)) \* p is packet *\
        Transmit (p(ID_f(S))) and En-queue (p(ID_f(S)))
        If {receive NACK && Loss = p(ID_f(S))}
            Re-transmit (p(ID_f(S))) and En-queue (p(ID_f(S)))
        else if {receive B-ACK}
            De-queue ( p(ID_f(S)))
    else if {n = Pd}
        If {no. of in-order received packet = ½ x Retransmit-buffer}
            Transmit (B-ACK)
        else if {loss detected}
            Transmit (NACK)
end for
```

III. Simulation Results

We have performed extensive simulation, and the performance of the protocol is compared with RCRT. As in Fig. 2, our work completely outperforms RCRT in terms of end-to-end retransmission delay.

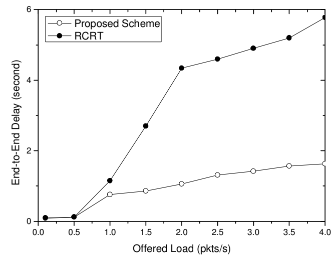

Fig. 2: End-to-End Delay Comparison

IV. Conclusion and Future Work

The proposed work is a semi-distributed reliable solution for WSN, where each retransmission is ensured through reserving the channel. In future we will extend this work to a complete data transport protocol for WSN.

References

[1] H. Zhang, A. Arora, Y.-r. Choi, and M. G. Gouda, “ Reliable bursty convergecast in wireless sensor networks,” in MobiHoc ’ 05: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing. New York, NY, USA: ACM, 2005, pp. 266– 276.

[2] C. yih Wan, A. T. Campbell, and L. Krishnamurthy, “ Pump slowly, fetch quickly (psfq): a reliable transport protocol for sensor networks,” in IEEE Journal on Selected Areas in Communications, 2005, pp. 862– 872.

[3] J. Paek and R. Govindan, “ Rcrt: rate-controlled reliable transport for wireless sensor networks,” in SenSys ’ 07: Proceedings of the 5th international conference on Embedded networked sensor systems. New York, NY, USA: ACM, 2007.

[4] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker, and I. Stoica, “ Flush: a reliable bulk transport protocol for multihop wireless networks,” in SenSys ’ 07: Proceedings of the 5th international conference on Embedded networked sensor systems. New York, NY, USA: ACM, 2007.