2010 10th Annual International Symposium on Applications and the Internet

Reservation Based Reliability Control in Wireless Sensor Networks

Md. Obaidur Rahman*, Muhammad Mostafa Monowar*, Choong Seon Hong*, and Young An Kim**
*Kyung Hee University, South Korea
**Korea National Defence University, South Korea
mdobaidurrahman@gmail.com, monowar@networking.khu.ac.kr, cshong@khu.ac.kr, and roundsun26@hotmail.com

***Abstract*—This paper presents a reservation based reliability control protocol for wireless sensor networks having the applications that demand in-order and 100% delivery of the generated packets. Unlike earlier works the proposed reliability control avoids channel contention and collision problem during retransmission, and ensures guaranteed packet delivery. Moreover, the reservation based retransmission can ensure the maximum achievable channel capacity at each hop taking the interference into account. The performances are measured through extensive simulations which establish the efficiency of proposed work in terms of delivery ratio and delay.**

***Index Terms*—Wireless sensor networks; Reliability; Retransmission; Channel utilization; Reservation based schedule;**

I. Introduction

Wireless sensor networks (WSNs) are emerged to support variety of applications and fulfill their requirements. Traditional WSN includes monitoring applications, where sensor nodes are deployed and operated only to detect any anticipated event. However, to successfully identify an event a desired reliability should be maintained, because in recent days there exists many applications (i.e., surveillance detection for security and battle field, seismic vibration variation in structural health monitoring etc.), which demand in-order and 100% delivery of the generated packets.

Moreover, WSNs are energy constraint and researches are going on to prolong the network life time. Many researches such as: wake-up and sleep scheduling [1], energy aware topology control, power control exist currently to maximize energy conservation for sensor networks. However, due to frequent wireless interference, medium contention and collision, it is very challenging to restrict the transmission failure and it is beyond the scope of the mentioned topics. Hence, from the energy point of view such transmission failure certainly costs more energy as retransmission takes place for the lost data.

In an analysis of [2], it is shown that the existing retransmission based reliability control techniques increase channel contentions and packet collisions due to the lack of appropriate retransmission schedule. Furthermore, despite the advantageous properties (i.e., energy saving) of NACK-based end-to-end retransmission, existing reliable protocol 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. However, we argue that a retransmitted packet should get more priority in medium access, and a reservation based solution can fulfill this demand.

In this paper, we motivate to design an end-to-end medium reservation schedule in order to retransmit any data. The proposed reliability control protocol aims to avoid wireless interference, contention and collisions during retransmission, providing higher priority to the retransmitted packet. The medium is reserved such a way that it can work for both single or multiple packet(s) retransmission while maximizes channel utilization at each hop. The performance of the proposed work are evaluated through extensive simulations.

In rest of the paper, Section II presents related work on existing reliable protocol of WSN. Section III and Section IV include preliminary design considerations and detail protocol operations respectively. Section V describes the simulation efforts and Section VI concludes the paper with future direction.

II. Related Work

Several works have been done over the years for the reliability control in WSN, and those can be categorized into *hop-by-hop* and *end-to-end* protocols.

The protocol PSFQ [4] is a hop-by-hop protocol, where an intermediate node stops data forwarding and sends a NACK to the previous hop when detects any packet loss. Data forwarding resumes when the retransmitted packet is received by the intermediate node. Thus, it increases delay for other packets and enlarge queue occupancy at intermediate nodes.

The end-to-end protocol RCRT [3] not only provides reliability but also control the congestion with a centralized approach. However, in this paper our primary concern is reliability control, and in Section I we have addressed the shortcoming of RCRT's reliability module.

The protocol Flush [5] also provides end-to-end reliability. But the operation of this protocol is limited to work on a linear topology only, and solves the intra-flow interference problem while the effects of inter-flow interference is not revealed in this protocol.

III. Preliminaries

A. System Model

We consider a sensor network having sensor nodes deployed within a bounded terrain. One or multiple nodes act as the source(s) which sends data packet to the sink through intermediate nodes of a multi-hop routing path. One or more sinks

978-0-7695-4107-5/10 $26.00 © 2010 IEEE
DOI 10.1109/SAINT.2010.58

may exist to receive that application data and sink node knows its hop distance from each node. We assume the network as a tree topology rooted at sink, where the *parent-child* relation can be defined as *downstream-upstream*. A flow is assumed as the traffic from a particular source node.

Let, there are N sensor nodes in the network and the i-th node is denoted as n_i, whereas the j-th source node is denoted as s_j. The proposed reliability control is independent of number of sink nodes, hence we simply select one node as final destination (sink) and use d to denote it. The source-sink and sink-source bidirectional routing paths are considered as $path_{(s_j,d)}$ and $path_{(d,s_j)}$ respectively. Each routing path comprises multiple intermediate nodes and the k-th intermediate node on these path(s) is denoted as i_k. The hop distance of a node (or the sink) from a particular source s_j is expressed as $h^{n_i}(s_j)$. Again, the neighbor node sets of s_j and i_k are expressed as $N(s_j)$ and $N(i_k)$ respectively. A flow from source s_j defined as $f(s_j)$.

B. Assumptions

The effect of interference is taken into account for the proposed protocol. In order to achieve better channel utilization in retransmission a pipelined data forwarding model is designed, where we denote the interference range as R_I. The motivation is to ensure the maximum one-fourth ($\frac{1}{4}$) of the total wireless capacity for a single hop transmission [6], and like [7] we assume $R_I = 2$, that is a node's any sort of transmission interferes its all the 2-hop neighbor's reception.

IV. Reservation Based Reliability Control

The proposed reliability control is independent of MAC layer retransmission, rather it uses an end-to-end approach to meet the purpose. Like RCRT [3], we use the end-to-end NACK-based solution to notify a source about the lost packet(s). However, in RCRT the end-to-end retransmission of any packet is done as normal transmission, whereas, for guaranteed delivery we make reservation based schedule for the retransmission and it is the fundamental difference between RCRT and our work. Furthermore, the end-to-end scheme has several advantages to make any reliability control protocol more robust. First, it avoids ACK explosion in the network which in turn saves energy; Second, it can ensure in-order delivery of the packets that is the stringent demand of many applications (i.e., surveillance detection, seismic variation).

A. Hop Differentiation and NACK Forwarding

At the beginning each node n_i measures its forwarding interval (denoted as T_i), which is the average medium access interval that includes the medium access delay and data transmission period. Using the exponential moving average formula, we compute T_i as:

$$T_i = (1 - \alpha) \cdot \widehat{T}_i + \alpha \cdot T_i \tag{1}$$

where, $\widehat{T}_i$ is the medium access interval of the last transmitted packet and α is the a smoothing constant, $0 \leq \alpha \leq 1$.

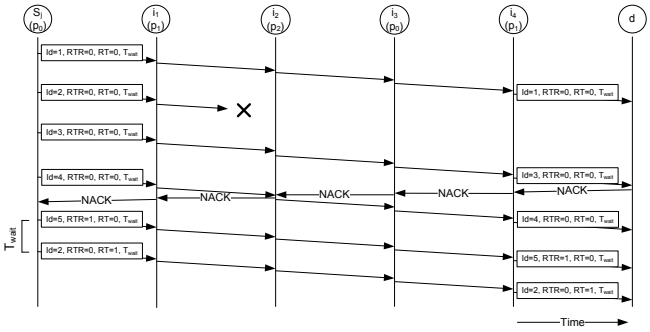

Fig. 1. Single packet retransmission. Upon receiving the packet (id=3), sink initiates the NACK for the lost packet (id=2). Source s_j sends the RTR request through the packet (id=5) and reserved the retransmission schedule after T_{wait}. Intermediate nodes (i_k) also forward the schedule and accordingly participate in the retransmission. The schedule allows s_j (for packet id =2) and i_4 (for packet id=5) to transmit simultaneously.

To design the protocol, we make a hop differentiation between the nodes based upon routing layer information. The sink extracts its hop distance ($h^{n_i}(s_j)$) from each source s_j. While forwarding the NACK on $path_{(d,s_j)}$, the $h^{n_i}(s_j)$ value is included in NACK header. Furthermore, upon receiving the NACK an immediate upstream node of the sink assigns its hop distance from s_j. Note that at each hop the $h^{n_i}(s_j)$ value is updated by subtracting one, and all the NACK forwarding nodes on $path_{(d,s_j)}$ do the same. Thus the source has its own $h^{n_i}(s_j)$ value equal to zero.

Furthermore, a hop differentiation metric is defined as $H(n_i)$, and all the NACK forwarders including the source and sink calculate their hop differentiation metric for s_j as (where, we use $R_I = 2$):

$$\underset{n_i \in path(d,s_j)}{H(n_i)} = h^{n_i}(s_j) \mod (R_I + 1) \tag{2}$$

Nodes on $path_{(d,s_j)}$ are categorized as p_0, p_1 and p_2 nodes for the $H(n_i)$ values 0, 1 and 2 respectively. Intuitively, nodes in every 3-hop from the source are the p_0 node(s) having 0 value for hop differentiation (as in Fig. 1).

Additionally, each of the NACK forwarder piggy-backs another field in NACK, named as *Maximum Forwarding Interval* (MFI). The NACK forwarders update the MFI field using the following rules:

- If the NACK forwarder is a p_0 node, then it simply discards the current MFI and sets its own forwarding interval (T_i) as the new MFI. However, a p_0 node stores the MFI obtained from downstream and uses it while retransmitting a packet.
- If the NACK forwarder is a p_1 or p_2 node and the current MFI in NACK is less than its forwarding interval (T_i), then it replaces the MFI with T_i, otherwise keeps it unchanged. We set zero for the MFI of sink.

The rules mentioned above creates the scope to maximize channel utilization by all the participating nodes in retransmission (discussed in Section IV-B).

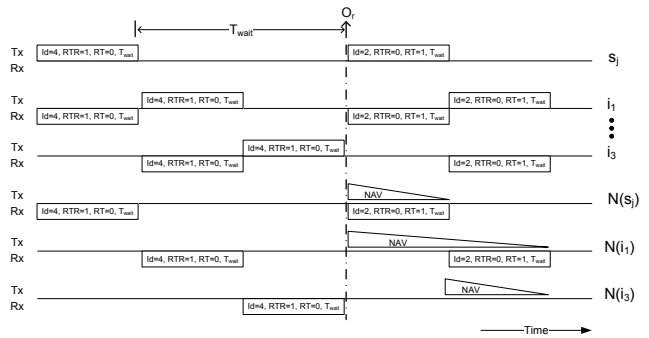

Fig. 2. Timing diagram NAV update by the neighbors of retransmitting nodes.

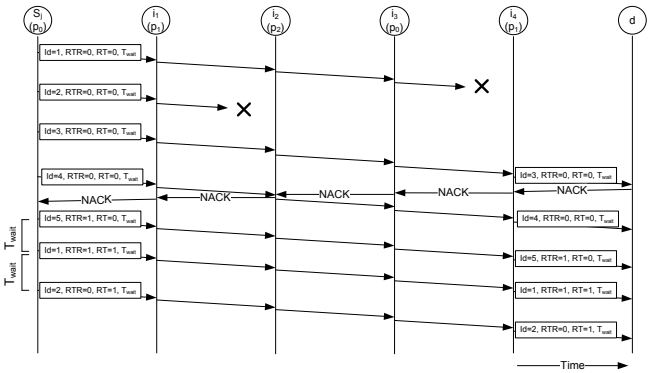

Fig. 3. Multiple packet retransmission. After the retransmission of the packet (id=1), source s_j injects another retransmission (id=2) after an updated T_{wait} interval. Here also s_j and i_4 transmits simultaneously.

B. Single Packet Retransmission

Upon receiving a packet, sink node d checks the sequence id of each flow $f(s_j)$, and whenever a gap is detected in sequence it issues a NACK and sends the lost packet retransmission request to the source s_j. Moreover, the sink uses a forwarding interval based estimation to detect any retransmission loss or NACK loss, and issues another NACK for the same lost packet(s). Here, we assume that the T_i value of each node on the routing path $path_{(s_j,d)}$ are provided to the sink.

On the contrary, source s_j stores a copy of all the transmitted packet in a retransmission queue, until it receives an explicit B-ACK (Block ACK) from the sink having the highest sequence id of the in-order received packets. It takes $h^{n_i}(s_j)$ successful transmission to propagate the B-ACK from sink to source, and whenever the number of in-order reception of a particular flow crosses half ($\frac{1}{2}$) of the retransmission queue, sink sends a B-ACK to the source. However, upon receiving a NACK having a single packet retransmission request, source s_j announces a channel reservation schedule in its next packet. Source's immediate downstream node receive the packet, and further broadcast the a reservation schedule such that it can receive the retransmitted packet first and immediately transmit the same packet to the downstream without any contention and delay. In the same way, the schedule for the same retransmission is announced by the nodes on $path_{(s_j,d)}$.

To reserve the wireless medium for retransmission we include the following fields in each packet header as in Fig. 1: one flag bit for the retransmission request (RTR), duration of the intended reserved period for sending or receiving or both (D_r), intended reserved period's start offset (O_r), and another flag bit RT to represent the status of the current packet (i.e., retransmission or not). If there is any packet to be retransmitted, the source sets 1 in RTR and announces O_r after a waiting period. We refer this waiting period as T_{wait}, and source s_j as well as all the downstream p_0 node (on $path_{(s_j,d)}$) calculate T_{wait} using their stored MFI (derived in Section IV-A):

$$T_{wait} = 3 \times MFI \tag{3}$$

Therefore, T_{wait} can ensure announcement of retransmission schedule (i.e., for sending or receiving or both) by the downstream node at 3-hops away from the source. Note that a downstream p_1 or p_2 node updates the T_{wait} by deducting the elapsed time till it receives the reservation schedule from the upstream source or a p_0 node. Since the neighbors of the nodes on path $path_{(s_j,d)}$ are well-aware about the reserved retransmission schedule, they allocate their *Network Allocation Vector* according to the D_r value; hence, a collision free retransmission can be anticipated for the lost data packet.

The proposed retransmission follows an almost staggered forwarding [8] as in Fig. 1. where nodes separated by 4-hops can transmit simultaneously, and ensure one-fourth ($\frac{1}{4}$) use of the total channel capacity at each hop. Fig. 2 shows the *Network Allocation Vector* settings of the neighbors of nodes on $path_{(s_j,d)}$.

C. Multiple Packet Retransmission

While the sink detects multiple packet loss of the same flow, it initiates the NACK with a list of lost packets. On the other hand, upon receiving that NACK the source and its downstream on $path_{(s_j,d)}$, perform the first packet retransmission in the similar way as described in Section IV-B. However, the difference comes while transmitting the next retransmission. To reserve the medium for the second retransmission source node makes some changes in the header of the first retransmitted packet: (i) It keeps 1 in the RTR bit, and (ii) Updates the T_{wait} period as:

$$T_{wait} = 3 \times T_{Data} \tag{4}$$

where, T_{Data} is the transmission time of a data packet at each hop. Therefore, the source injects the next retransmission when the first retransmitted packet have traversed at least 4-hops from it. Again it is assured here that the 4-hops distant nodes can transmit simultaneously and achieve the maximum channel utilization. As shown in Fig. 3, after retransmitting the packet (id=1), source injects another retransmission (id=2) just after the updated T_{wait} time.

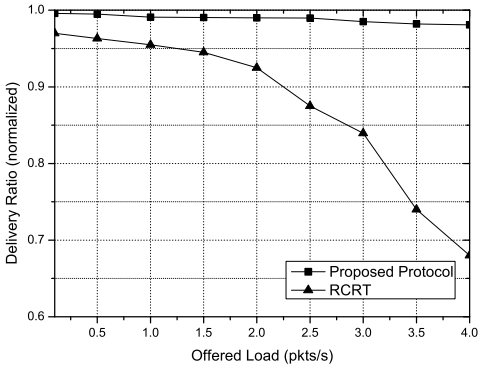

Fig. 4. Normalized delivery ratio.

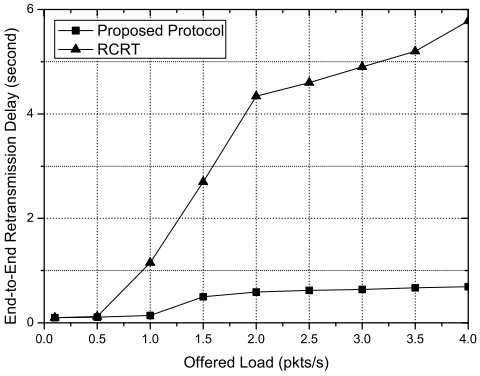

Fig. 5. End-to-end retransmission delay.

V. Evaluation

A. Simulation Setup

We have performed extensive simulations in NS-2 simulator for a tree topology having a single sink. A network of $100m \times 100m$ area is used for the topology, where 100 nodes are deployed in a uniform random distribution. The paths of the tree are pre-computed and routing traffics are avoided in the simulation. The channel bandwidth is set as 250 *Kbps*. We set $30m$ of transmission range and $60m$ for carrier sensing range for the sensor nodes. The payload size of the data packets are 32 bytes. It is considered that 15 nodes in randomly selected locations of the network generate data with a rate varies from 0.1 to 4 pkts/s. Furthermore, the NS-2 implementation of the IEEE 802.11 is used as underlying MAC protocol. The simulation is executed for 200 seconds and we have taken 10 simulation executions.

The performance of the proposed protocol are compared with the RCRT protocol using the following metrics:

- *Delivery Ratio*: It is the ratio between total number of packets received by the sink to the total number of packets generated by the source nodes.
- *End-to-End Retransmission Delay*: The time gap between a retransmission starts at source and received at sink.

B. Simulation Results

In Fig. 4 the normalized delivery ratio of both RCRT and our proposed protocol are plotted. According to the result shown in the graph, our proposed protocol clearly outperforms RCRT having 99% delivery ratio. Although in low data rates RCRT have a better delivery ratio, but as the rate increases it falls sharply. The results shows the effectiveness of the reservation based retransmission in proposed protocol; on the contrary, the result of RCRT suffers due to contention and collision in normal transmission.

We observe in Fig. 5 that in compare to RCRT our scheme maintains a stable and lower end-to-end delay for the retransmitted packet. For RCRT the delay increases dramatically with the increment of the traffic load; in contrast, due to the reservation based staggered and pipelined data transmission the proposed protocol achieves a better delay performance.

VI. Conclusion

In this paper, we propose an efficient reservation based reliability control protocol for WSNs. The simulation results shows the effectiveness of the work over RCRT in terms of packet delivery ratio and end-to-end delay of the retransmitted packets. We believe the proposed protocol can be extended in a larger extent for a complete data transport protocol and we leave this task as our future work.

Acknowledgment

"This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency)" (NIPA-2010-(C1090-1021-0003)). Dr. C.S. Hong is the corresponding author.

References

[1] W. Ye, J. Heidemann, and D. Estrin, "Medium access control with coordinated adaptive sleeping for wireless sensor networks," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 493–506, 2004.

[2] 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.

[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, pp. 305–319.

[4] 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.

[5] 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, pp. 351–365.

[6] J. Li, C. Blake, D. S. De Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*. New York, NY, USA: ACM, 2001, pp. 61–69.

[7] H. Zhai and Y. Fang, "Distributed flow control and medium access in multihop ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 11, pp. 1503–1514, 2006.

[8] G. Lu, B. Krishnamachari, and C. S. Raghavendra, "An adaptive energy-efficient and low-latency mac for tree-based data gathering in sensor networks: Research articles," *Wirel. Commun. Mob. Comput.*, vol. 7, no. 7, pp. 863–875, 2007.