Congestion Detection and Control Algorithms for Multipath Data Forwarding in Sensor Networks

Md. Abdur Razzaque and Choong Seon Hong*
Department of Computer Engineering, Kyung Hee University (Global Campus), South Korea
E-mail: m_a_razzaque@yahoo.com, cshong@khu.ac.kr

Abstract — **Recently, there have been a number of research works on multipath data forwarding mechanisms that can increase the source data transmission throughput, data security and enhance the network lifetime. However, a congestion control mechanism dedicated for multipath routing is rarely found in the literature. This work first proposes a source data packet loading scheme over multiple paths and then presents congestion detection and control algorithms suited for multipath data forwarding. Our congestion detection is based on buffer occupancy of a node and the control algorithm is driven by preconceived packet loading rates maintained at each source node. The simulation results show that our schemes exhibit better performance than the existing ones.**

Keywords — **Wireless Sensor Networks, Multipath Data Forwarding, Congestion Control, Packet Delivery Ratio, End-to-End Throughput.**

1. Introduction

A typical Wireless Sensor Network (WSN) consists of a base station (sink node) and a large number of tiny sensor nodes. The nodes are characterized by limited energy supply, reduced computation capability, less bandwidth and susceptible to failure [1][2]. When a sensor node detects some phenomena, happened in the network, it forwards data packets towards the sink. If these data packets coming from a larger group of sensor nodes are routed over the shortest paths, an onset of congestion is happened in the middle or near the sink node of the network. This happening is attributed either by buffer overflow or by high degree of media contention [3]. Whatever the reason of congestion is, it causes the reduction of node throughput and packet delivery ratio. It also increases the energy wastage of the nodes which in turn decreases the network lifetime [3][8].

Recently, there have been a number of research works attempting to increase the sensor node data transmission throughput, packet delivery ratio and data security via multipath routing [4][5][6]. In [4], interference-aware multiple paths between source and destination are established to split the source traffic across the paths. In [5], multiple copies of a data packet are sent along different paths, allowing for resilience to failure of certain number of paths. In [6], multipath routing is used to rapidly find alternate paths between sources and sink and in [7], it is used to increase the security of data packets sent from the source sensors. Therefore, multipath routing has been proved a promising technique to achieve a number of advantages in WSN.

While a greater number of research works are available in the literature [2], [3], [8] addressing the problem of congestion detection and control in single path (or shortest path) routing paradigm, it is rarely found any algorithm suited for multipath paradigm. PCCP [2] is a traffic priority based congestion control protocol, it is primarily designed for single path routing network and then an extension has been given for multipath case. In PCCP, the incipient congestion is detected using the ratio of packet service time to the packet inter-arrival time at a node. When this ratio becomes greater than 1, congestion is detected. It does not care about the current status of the node buffer. Hence, false congestion may be detected frequently due to temporary unfairness of the underlying MAC protocol. In [8], a congestion avoidance protocol based on lightweight buffer management in wireless sensor network is proposed, where a sensor node is allowed to send a packet to its downstream neighbor only when the latter has buffer space to hold the packet. This scheme uses 1/6-buffer algorithm to solve hidden-terminal problem, i.e., every sensor advertises only one sixth of its remaining buffers. Even though it can significantly minimize the packet drops due to buffer overflow, the buffer utilization is very low. Moreover, in no way it is applicable for multipath data forwarding since it is assumed in [8] that all the upstream nodes of a downstream always send their data packets to the latter.

In this paper, we first propose a simple but efficient scheme for packet loading rate along the multiple paths from a source. The intermediate nodes along the paths monitor and updates the average size of local buffer (i.e., number of data packets in the buffer) using exponential weighted moving average and thereby detects an incipient congestion. It helps a node to increase the congestion detection accuracy or in other words, it minimizes the chance of false congestion detection. The proposed congestion control algorithm takes the presence of multiple paths into consideration while adjusting the packet loading rates along them. A number of predefined packet loading rates are considered for rate reductions in the event of network congestion. The performance of the proposed schemes are evaluated and compared with other mechanisms in *ns-2* [9] and the results show that our schemes provide much better performance than the existing ones.

The rest of the paper is organized as follows. Section 2 describes the network model and multipath routing and section 3 details the proposed schemes and algorithms. The

This research was supported by the MKE under the ITRC support program supervised by the IITA" (IITA-2008-(C1090-0801-0002))
* Dr. Choong Seon Hong is the corresponding author.

performance evaluation is done in section 4 and the section 5 concludes the paper.

2. Network Model and Multipath Routing

We consider a sink-rooted tree-based network architecture, where a large number of tiny sensor nodes uniformly distributed over the terrain. All sensor nodes have the same transmission and sensing range. Additionally, these sensor nodes have limited processing, power, storage and energy; while the sink node has powerful resources to perform any tasks or communicate with ordinary sensor nodes. We assume that sizes of the data packets transmitted by different nodes are same. The buffer size of a node is counted as the maximum number of data packets that it can hold. We also assume that CSMA/CA MAC is operating at layer 2 of each node.

The underlying routing layer protocol uses multipath data forwarding. Therefore, each source node has to establish multiple paths to the destination using any of the available multipath routing algorithms [4], [6], [7] and [10]. We choose to establish three alternate paths, namely P1, P2 and P3. The source node uses P1 and P2 concurrently for splitting the traffic load along these paths and switches to the third path P3 only when either one fails. Thus it minimizes the route reestablishment costs. The reason why we choose only two paths for load balancing rather than using all three is experimented and commented out in the results of [10]. There is no or very little advantage of using more than two paths in terms of throughput maximization.

3. Proposed Algorithms

The basic idea of the proposed mechanism is stated as follows. At first a source node starts to send data packets over two different paths at a predefined rate. The congestion detection algorithm at each intermediate node is invoked at reception of every data packet. If congestion is detected, a congestion notification packet is sent back to the source and then the source invokes the proposed congestion control algorithm to readjust the packet loading rate.

3.1 Source Packet Loading Rate (PLR) Initialization

As discussed in Section 2, each source node knows the addresses of next hop nodes for each path created by the routing protocol. Whenever a node senses some physical phenomena, happened in the network, it starts to send data packets over the P1-P2 path pair at initial packet loading rate 1 (PLR 1), as shown in Fig. 1. For PLR 1, the source first transmits a packet on P1 and keeps two packet transmission time intervals blank. Then it transmits the next packet on P2, followed by another packet transmission on P1. Thus the source node loads each path with one packet for every three packet transmission time intervals, i.e., the packet loading rate is at 1/3 of link data rate per path. Similarly, the source node maintains three other predefined PLRs, namely PLR 2, PLR 3 and PLR 4 and they are corresponding to 1/4, 1/6 and 1/8 of link data rate per path, respectively. Note here that these PLRs will be invoked only if the network is detected with congestive state and the selection of a particular PLR is determined by the congestion control algorithm, to be discussed in Section 3.3.

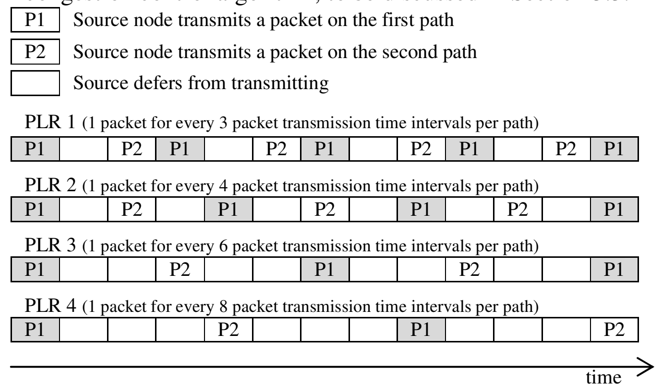

Figure 1. Preconcieved Packet Loading Rates of a source node

3.2 Congestion Detection Algorithm

The algorithm in Fig. 2 states the proposed congestion detection and notification strategies. Our congestion detection algorithm is buffer based. On reception of a data packet, each intermediate node monitors its current buffer size (BS) and calculates a running average value using exponential weighted moving average (EWMA) formula (line 10). If this average value becomes greater than a predefined threshold (Th) then the congestion is detected (see lines 11 to 14). Here, *w* represents the weight factor given to the current size of the buffer.

Once the congestion is detected, the intermediate node purges its buffer of all pending data packets in order to reduce the amount of backlogged packets (line 3). This also boosts up the forwarding of current data packet and it would be routed with minimum delay. After then, the node sends a congestion notification packet towards to source and sets a flag, which is cleared after a predefined time interval (lines 4 to 5). Before sending the current data packet to a designated next hop node, the congested node resets its EWMA variable.

```
1   /* flag(i): congestion notification packet flag */
2   If(CongestionDetection() && flag == FALSE){
3           PurgeBufferedPackets(); /* purges pending data packets */
4           SendCgstPacket(source); /*Congestion packet is sent to source*/
5           SetTimerForFlag(duration); /*Time duration to clear flag */
6           ResetEWMA(); /* reset EWMA variable to zero value */
7   }
8   SendDataPacket(next hop); /*data packet is routed to downstream */
9   CongestionDetection() {
10      BS_avg = (1-w) * BS_avg + w * BS_cur
11      If(BS_avg > Th)
12          return TRUE;  /* Congestion is detected */
13      Else
14          Return FALSE; /* No congestion */
15  }
```
Figure 2. Congestion Detection Algorithm

3.3 Congestion Control Algorithm

Whenever the source node receives the congestion control (CONGEST) packet sent by the congested node, it executes the congestion control algorithm, presented in Fig. 3. At first, the source node stops the forwarding of packets over the active paths. Then it reduces the PLR to the next higher rate. Before resuming the transmission of data packets at this new reduced rate, the source node sets a timer for the duration at which this new rate will be activated. If the source node does not receive any CONGEST packet during this period, it will switch to the next higher PLR. If the link qualities of any of the active paths deteriorate, eventually the source node starts to load at the lowest possible rate over that path. In the event of further deterioration of that path quality, the source node will receive more CONGEST packets but it has no option to decrease the PLR further. In this case, the source attempts to switch the congested path with the backup path if possible. Otherwise, the source reinitiates path discovery again.

```
If(received the congestion notification packet) {
      StopLoadingPackets();
      SetWaitingTimer(delay); /*wait for a certain amount of time*/
      switch(PLR_cur) {/*current packet loading rate*/
           Case PLR 1: {PLR_cur = PLR 2; break;}
           Case PLR 2: {PLR_cur = PLR 3; break;}
           Case PLR 3: {PLR_cur = PLR 4; break;}
           Case PLR 4: {break;}
      }
resumePacketLoading(delay,PLR_cur); /*starts transmit in new rate*/
```

Figure 3. Congestion Control Algorithm

4. Performance Evaluation

The effectiveness of the proposed schemes is evaluated through packet level simulation of traffic flows in ns-2.30[9]. We have implemented our schemes as well as PCCP[2] and LWBM[8] in order the compare the performances in terms of the following two metrics:

End-to-End Throughput: We measure the amount of data bytes received by the sink per unit time to calculate the throughput. In the graph, we put this value in Kbps for different traffic loads. We do not consider any control packets in this calculation. The higher the value is, better the performance is.

Packet Delivery Ratio: It is the ratio of the total number of packets received by the sink node over the number of packets sent by all source nodes. We express it in percentage, the higher the value is, better the performance shown by the mechanism.

The simulation parameters are described as follows. The terrain size is 1000mX1000m, where 1000 sensor nodes are uniformly distributed. The sink node is located at [1000, 500] location. Each sensor's transmission and sensing radii are 100m and 50m, respectively. The payload amount of each data packet is 64 bytes and the buffer size of each node is 30. The link data rate is 512Kbps. 100 sensor nodes are randomly picked as data source and the simulation is run for 10 seconds.

As the results presented in Fig. 4, at increased packet generation rates, the end-to-end throughput observed by the proposed mechanism is at least 1.5 times higher than the existing approaches. Also, the packet delivery ratio is very high as compared to existing schemes. This is due to multipath packet loading and rate adjustment policy, employed in the proposed scheme.

5. Conclusion

The proposed congestion control mechanism is able to increase the end-to-end throughput by loading the active paths at the highest possible rate than can be supported. We would like to extend this work attempting to theoretically analyze the accuracy and energy efficiency of congestion control mechanisms in wireless sensor networks.

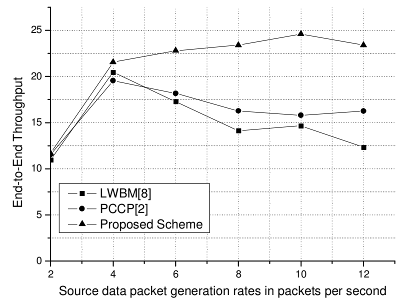

Figure 4. Throughput vs. Packet generation rates

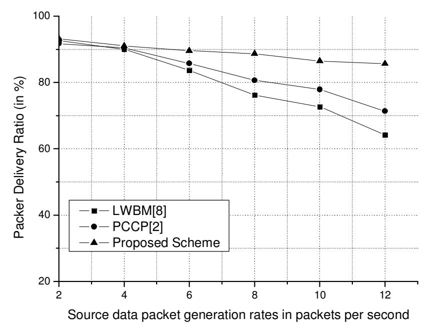

Figure 5. Packet Delivery Ratio vs. Packet generation rates

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayici, "A Survey on Sensor Networks", IEEE Commun. Mag., Vol. 40, No. 8, pp.102-114, 2002.

[2] Wang C, B. Li, K. Sohraby, M. Daneshmand, Y. Hu, "Upstream Congestion Control in Wireless Sensor Networks Through Cross-Layer Optimization", Vol. 25, No. 4, pp.786-795, 2007.

[3] C.Y. Wan, S.B. Eisemann, and A.T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks", Proc. ACM SenSys, pp.266-279, Los Angeles, Nov. 2003.

[4] J.Y. Teo, Y. Ha, C.K. Tham, "Interference-Minimized Multipath Routing with Congestion Control in Wireless Sensor Network for High-Rate Streaming", IEEE Trans. Mobile Computing, Vol. 7, No. 9, pp.1124-1137, Sep. 2008.

[5] M. A. Razzaque, M. M. Alam, M.M. Rashid, and C.S. Hong, "MCQoS: Multi-Constrained QoS Geographic Routing for Heterogeneous Traffic in Wireless Sensor Networks", IEICE Trans. Comm., Vol. E-91B, No. 8, pp.2589-2601, Aug. 2008.

[6] D. Ganesan, R. Govindan, S. Shenker and D. Estrin, "Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks", ACM Mobile Computing and Communications Review, Vol. 5, No. 4, pp. 10-124, Oct. 2001.

[7] F. Lu, L. Geng, L. Chia, Y. Liang, "Secure Multi-path in Sensor Networks", Proc. 5th Int. Conf. Embedded Networked Sensor Systems, pp.413-414, 2007.

[8] S. Chen, and N. Yang, "Congestion Avoidance based on Lightweight Buffer Management in Sensor Networks", Proc. IEEE Trans. on Parallel and Distributed Systems, Vol. 17, No. 9, pp.934-946, 2006.

[9] The Network Simulator – ns-2.30, http//www.isi.edu/nsnam/ns/ns-build.html.

[10] E.P.C. Jones, M. Karsten, and P.A.S. Ward, "Multipath Load Balancing in Multi-Hop Wirless Networks", Proc. IEEE Intl. Conf. Wireless and Mobile Computing, Networking and Comm. (WiMob' 05.), vol. 2, pp. 158-166, 2005.