Congestion Aware Fair Data Delivery in Wireless Multimedia Sensor Networks

Fernaz Narin Nur, Selina Sharmin,
Md. Abdur Razzaque
Department of Computer Science & Engg.
University of Dhaka, Dhaka 1000, Bangladesh
farnaz_amin, selina_sharmin@yahoo.com
razzaque@cse.univdhaka.edu

M. Abdullah-Al-Wadud
Department of Industrial and
Management Engineering
Hankuk University of Foreign Studies
Gyeonggi-do, South Korea
wadud@hufs.ac.kr

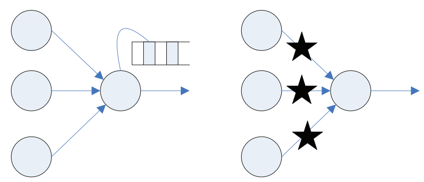
Abstract: In this paper we present a novel energy efficient congestion control scheme for Wireless Multimedia Sensor Network called, CFD (Congestion aware Fair Data delivery). In our proposed CFD mechanism both node level and link level congestions are detected. It ensures fairness of each packet when packet drops happens due to congestion. To avoid congestion a fair rate allocation mechanism is also proposed. CFD has been evaluated extensively using simulations, and the results have shown that CFD provides a better performance with minimum delay than those of existing approaches.

Key–Words: Mutlimedia Sensor Networks, Congestion Control, Fair Data Delivery, Rate Control, End-to-end Packet Delay

1 Introduction

A Wireless Multimedia Sensor Network (WMSN) consists of spatially distributed autonomous sensor nodes for monitoring physical, environmental, real-time event tracking or health care conditions for example, temperature, sound, pressure, light etc. Advancements made in sensing technology, source coding techniques, and availability of low cost CMOS cameras have made WMSN a reality. WMSN contains sensor nodes which have audio or both audio and video sensing capabilities. The WMSNs applications are capable of generating real-time as well as non real-time data traffic. Real-time multimedia applications require lower bounds on delay and jitter along with some bandwidth guarantees. Furthermore, critical event data requires higher reliability and lower delay. On the other hand, non real-time applications do not have stringent Quality-of-Service requirements in terms of bandwidth, delay and jitter.

The event-based nature of WMSNs leads to unpredictable network load. The detected event normally generates a bursty traffic from many sensor nodes simultiniously. This sudden surge of traffic converges at somewhere near the sink, which can result congestion in the network. Congestion may happen in WMSN due to occurrence of a critical event, excessive event reporting, multimedia data, and hot spots etc. The warning sign of congestion in sensor network is the raise in buffer drop rate and packet delay, degradation of radio channel quality and network throughput. Also the network gets biased towards delivering data from nodes closer to the base station and repulsively unfair towards nodes located further away from the sink.



(a) Node level congestion (b) Link level congestion

Figure 1: Congestion types

In WMSN congestion may be caused due to two reasons: the first one is the node level congestion which occurs when the packet-arrival rate exceeds the packet-service rate causing buffer overflow in the node, as shown in Fig 1(a). The second one is the link level congestion which can take place due to excessive media contention, interference for the shared medium since many source and routing nodes try to access the media simultaneously as shown in Fig.1(b). This congestive condition increases packet losses, queuing delay. For judicious and accurate detection of congestive states both the causes should be taken care into consideration.

Congestion detection and avoidance and control mechanisms are necessary to meet QoS requirements.

ISBN: 978-960-474-361-2

Additionally, it is highly desirable to provide fairness among different flows. Fairness is concerned with the relative throughput of the flows sharing a link. When packet drops occur for congestion, it is not desiarable to treat all packets equally. Because in WMSN some critical or vital packets may require high reliability and no dealy on the other side some may need only reliability. So faireness of each packets should be required to differentiate among different classes of packets.

In the literature, extensive works have been done to address congestion problems [1]-[2]. A group of works designed for reliable end-to-end data delivery from every sensor to a sink [1][3] and other group of works designed for hop by hop congestion control at every intermediate node in the network from source to sink [2][4]. For example to detect congestion, IFRC[5] uses queue size and shares this congestion states among others. But IFRC[5] is less flexible as it uses sophisticated tuning parameters. ECODA[6] has detected node level congestion but link level congestion is not well addressed, different data traffic are prioritized skillfully but static priority is not well distinct and as dynamic priority is closely related to static priority, so proper fairness is not given to route through traffic. In our proposed CFD mechanism both node level and link level congestions are detected. It will ensure fairness of each packet when packet drops happens due to congestion and to avoide congestion a fair rate allocation mechanism is proposed. The proposed work integrate the following key ideas:

- Uses dual-buffer thresholds and buffer difference for node level congestion detection and rate control.
- Re-transmission mechanism is used for link-level congestion.
- A flexible queue scheduler can dynamically select the next packet for sending towards the next hop. Moreover, a novel technique is adopted to filter packets when congestion happens. A Queuing model for scheduling is used for this purpose
- An AIMD(Additive Increase Multiplicative Decrease) Rate control algorithm mechanism is proposed to efficiently handling the Rate of each node.

The rest of this paper is organized as follows. We describe related works and motivation in Section 2 and Network model and assumptions in Section 3. Our proposed mechanism is presented in Section 4 and the simulation results are presented in Section 5. Finally, we conclude the paper in 6 along with future research direction.

2 Related Works

A number of previous works have addressed the issue of congestion control in wireless sensor networks [7]. RMST[8] provide hop by hop reliable transport protocol that is spaecially designed to run on top of dirrected diffusion. Here packet loss is recovered using caches in the intermediate nodes in a hop by hop manner. RMST[8] provide reliability but the node's transmit rate is set by a system administrator and is designed for more capable sensor nodes. Another protocol ESRT [3] utilize centralized congestion control schemes. It adjusts source packet data sending rate by classifying the network into five regions. The ESRT's [3] centrally calculate the rate and cannot deal with transient congestion properly.

The first detailed investigation of congestion control in sensor networks was presented in Congestion Detection and avoidance CODA[1]. To detect congstion it sample the load of the medium as well as monitor the queue occupancy. Detecting congestion a node broadcasts a backpressure message to upstream nodes and the upstream nodes changes the traffic volume to reduce congestion. But in CODA[1] it doesn't explictly focus on per source fairness. The CCF [9] detects congestion in each intermediate sensor node by comparing packet service time with the available service rate. Congestion information is implicitly reported. To control congestion CCF use hop-by-hop manner and each node adjust rates based on its available service rate and child node number. CCF provides simple fairness which ensure each node receives the same throughput. However CCF depends only on packet service time for the rate adjustment that could lead to low utilization. Because some sensor nodes may not have enough traffic to send or there is some sort of significant packet error rate (PER). A node-priority based congestion control mechanism, PCCP [10] has been proposed for WSN. PCCP[10] detects congestion using ratio of packet service time with packet inter arrival time. PCCP[10] senses both node and link level congestion. However, it doesnt properly handle prioritized heterogeneous traffic in the network. Also sometimes false congestion may observed as it doesn't use the current status of the buffer.

RCRT[2] assures complete reliability using centralized congestion and rate control mechanisms. But it depends on MAC retransmission for one hop reliability and use end to end retransmission to recover loss packets. In IFRC[5] every node use multi-level buffer thresholds. IFRC try to ensure fairness for every node.

The PHTCCP[11] protocol provides rate control for prioritized heterogenious traffic. Intra-queue and inter-queue are used here to ensure feasible transmission rate among heterogeneous traffic. Also dynamic

ISBN: 978-960-474-361-2

transmission rate adjustment is used for efficiently utilize the link. But when congestion happens their rate adjustment give all child nodes to same rate that may not be feasible in all situation. The ECODA[6] protocol achieves fairness through Flexible Queue Scheduler. There are two sub-queues maintained with each node: one for local generated traffic and other for route-through traffic. In the route-through traffic, the packets are grouped by the source and arranged by their dynamic priority. The packets from both of the queue sent alternatively and for sending packets from the queue of route-through traffic, roundrobin policy is used.

The existing congestion control protocols for WSNs have two primary limitations. First, they only guarantee simple fairness, which means that the sink receives the same throughput from all nodes. However, sensor nodes may have different priority or importance due to either their functions or the location at which they are deployed. Second, most protocols fail to properly detect both link level and node level congestions. In our proposed work we detect both link level and node level congestion, and use a fair rate control algorithm to control the rate of nodes that will lead to a steady state situation.

3 Network Model and Assumptions

This section states the design considerations, network model, and preliminaries for the protocol which we have taken into account.

3.1 Network Model

We consider a wireless multimedia sensor network where many multi-purpose sensors are deployed over a specific target area. Each sensor node sends the sensed data packet to a single base station or sink. The nodes can also route data traffic originated by other nodes. Therefore, each node can act both as a source and a router. Data packets route through using many-to-one multihop single path routing as in Figure 2. The data traffic sending towards the sink can be either soft real-time or hard real-time or non real-time based on their applications. The sink and sensor nodes are aware of their geographic location information either via GPS (Global Positioning System) or any other location determination technique. Using some neighbourhood protocols (like [12]) each sensor node has knowledge of it's neighbors. We assume that the sensors are static after deployment, and the topology of the network does not change regularly. All nodes use a CSMA (Carrier Sense Multiple Access) based MAC protocol for media contention. The sensor nodes are supposed to send data periiodically or on the detection of an event or a combination of both. Here, some definitions have given based on which we have designed our protocol.

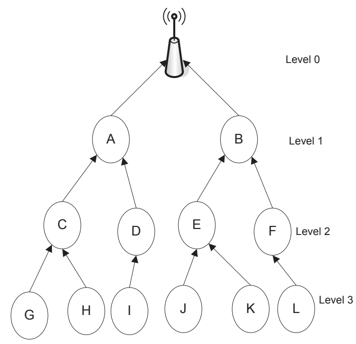


Figure 2: Network model

Definition 1 *A packet may contain Real-time or Non Real-time data traffic. Again Real-time data traffic can be either Soft Real-time or Hard Real-time. There is some message which should be delivered within a deadline and if the message arrives after its deadline, it is considered the failure of the system. This type of data traffic is considered as Hard Real-time traffic class. So, Hard Real-time data traffic should ensure deterministic end-to-end delay bound. While some message has no deadlines to deliver for which a probabilistic guarantee is required and some lateness is acceptable. Considering these factors, each packet is given a static priority, SP, based on its traffic class. Table 1 shows the static priorities of packets in heterogeneous environment where each node can contain different classes of packets.*

Packets	Non real time = 1	Soft Real time = 2	Hard Real time = 3	*SP(l)*
1	1	0	0	1
2	0	1	0	2
3	0	0	1	3

Table 1: Static priority for each packet

Definition 2 *The dynamic priority of each packet is calculated as*

$$DP(l) = DP(l+1) + \frac{hop}{T_{life}}, \qquad (1)$$

where, $DP(l)$ represents the dynamic priority of the packet in current node, $DP(l+1)$ represents the dynamic priority of that packet in the previous node, T_{life} is the remaining lifetime of the packet, hop is the number of hops from current node to the sink. The

ISBN: 978-960-474-361-2

DP of each packet is varied from node to node. When a packet has generated it's life time T_{life} value has set to indicate how long it will stay. So packets with lower T_{life} value need to give more priority as its expiration time is very short which we included here to calculate the DP of a packet.

Definition 3 *Global Priority of each packet is calculated as:*

$$GP(l) = SP(l) + DP(l) \tag{2}$$

The GP,DP and SP vale is attached in the forwarded packet so the downstrem nodes will get the values.

Global priority for each node is calculated as:

$$GP(i) = \sum_{j=1}^{n} GP(packet)_j \tag{3}$$

Definition 4 *The service rate R^i_{svc} is the rate of forwarded packets to downstream nodes by node i. The incoming rate R^i_{in} is incoming packet's rate. Here the incoming rate will be the summation of all the childrens outgoing rate plus the source packets generation rate of source node i. This can be stated as:*

$$R^i_{in} = \sum_{0}^{u} R^{i,u}_{svc} + R^i_{src} \tag{4}$$

Here $R^{i,u}_{svc}$ are service rate of i nodes upstream children nodes and R^i_{src} is the packet generation rate of source node i.

Definition 5 *The Buffer Rate difference for each node is calculated as:*

$$\delta r = R^i_{BO} - R^i_{BI} \tag{5}$$

This value is compared with the threshold value for detecting congestion.

Definition 6 *Control interval is the period of time over which a node takes a control decision regarding the increase or decrease of the transmission rate. In our implementation this interval can be a constant time like after each 30 ms the congestion state will be checked for any change of transmission rate.*

4 CFD design

The detail of proposed protocol is discussed in this section. The system architecture of the proposed work is given in figure 3. The Congestion Detection Unit (CDU) detects the congestion using buffer threshold value and retransmission value. In our procol, we avoid congestion depending on the level of congestion. Then, with the help of the CFD rate control algorithm each node allocates a new rate according to it's global priority. All the child nodes of a parent node overhear the congestion level and new rate of their parents using the broadcast characteristics of wireless and depending upon this they also allocate their new service rate.

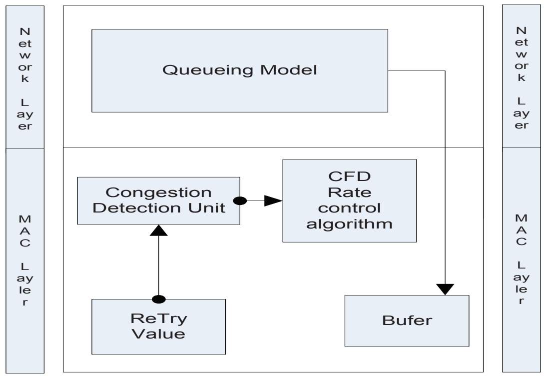


Figure 3: Network Architecture

4.1 *Queuing Model*

All packets generated by the sensor nodes do not have equal imporatnce. In the network different packets have different priority. So when congestion happens it is not feasible to drop packets arbitrary. It is desiarable to give more priority to vital packets so that they can reach the sink befor the normal packets. In the literature a few works has focused on this. ECODA[6] defines priotiry on packets but their priority is not well defined. In our protocol we use a classifier that will classify the packets according to their global priority and put them in the appropriate queue. When packets generate or come from child nodes their GP is calculated. The classifier will set the global priority of packets and classify the three classes of packets for both route through traffic and source traffic. The packets are put into three differnet queue. A schedular takes the three classes of packets, run a priority algorithm that will always take the high priority packets from the queue and put them in a buffer according to their GP. Packets are taken from this buffer and pass to the mac layer for sending. The GP of each packet is calculated in such a way that packets have lower life time has higher GP so it can be said that high priority packets will send first that will ensure the fairness of packets.

ISBN: 978-960-474-361-2

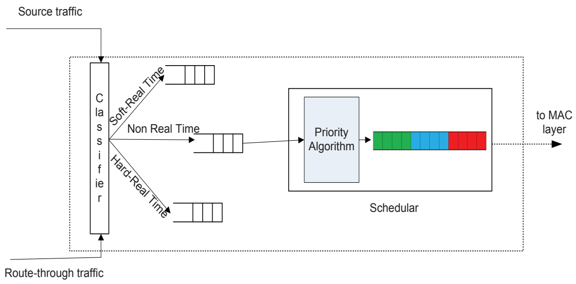


Figure 4: Queuing Model

4.2 *Use dual-buffer thresholds*

The buffer in the schedular is used to detect node level congestion here. Detecting node level congestion using the buffer is easy as this takes almost no cost. In the proposed protocol we have used two thresholds to detect congestion. Using this method, congestion level could be detected efficiently and fairness could be ensured.

Depending on the current status of buffer occupancy the states of buffer may reside in three states: accept state, filter state and reject state. Two thresholds named Q_{min} and Q_{max} are used to border different buffer states. If Q_T is the total buffer length then the two threshold values can calculated as:

$$Q_{min} = \frac{1}{3} \times Q_T \tag{6}$$

$$Q_{max} = Q_T - Q_{min} \tag{7}$$

The buffer changes its states according to different threshold values as in figure. A node takes its current buffer occupancy Q_{cur} and calculates its average buffer occupancy Q_{avg}.

$$Q_{avg} = (1 - \alpha)Q_{avg}^{prv} + \alpha Q_{cur} \tag{8}$$

Here the value Q_{avg}^{prv} is the previous average buffer value. The value α is the average moving coefficient and we set this value to 0.1. Different buffer states reflect different types of channel loading and based on that corresponding strategy is adopted to accept or reject packets in different states. In the "Accept State" all packets are allowed to transmit, in the "Filter State" some low priority packet will be dropped and in the "Reject State" not all packets but most of the packets will be rejected because buffer utilization is too high.

Depending on the buffer threshold values different actions will be taken for handling packets:

1. If $Q_{avg} < Q_{min}$ then all incoming packets are buffered to utilize the buffer

2. If $Q_{min} < Q_{avg} < Q_{max}$ then some low priority packets will be dropped or overwritten with some high priority packets.

3. $Q_{avg} > Q_{max}$ then some high priority packets from low priority node will be dropped or overwritten and all the upstream node's rate will be decreased along with its own rate.

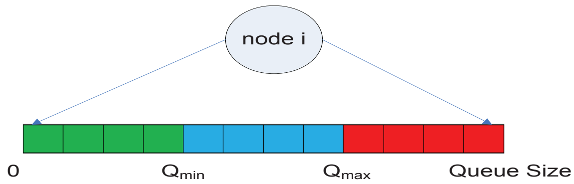


Figure 5: Buffer States

4.3 *Link-level congestion detection*

Many-to-one traffic flows in sensor network might converge somewhere near the sink node [13] and nodes in that area may be overloaded. This happens frequently as two or more nodes try to send data at the same time. Afterwhile the nodes sorrounding in the overloaded node would quickly be overloaded as well [13]. Although the buffer of that node is not fully occupied at congestion time but very soon it will be filled as data can't be forwarded. For link-level congestion the packet retransmission increases as packet experiences an unsuccessful transmission attempt more frequently. So, to measure link-level congestion we use the retry field of a packet whose value is 1 when it is a retransmitted packet. Using this we calculate the retransmission value RT_{val} as:

$$RT_{val} = \frac{\sum_{j=1}^{p} F_j}{P} \tag{9}$$

where j = the j^{th} packet, P = P packets retry value is used , F= Retry field which will be 0 or 1. Here P pakets retry value is checked to calculate the RT_{val}. We take the current measurment $RT_{val,cur}$ and find the adaptive estimation that reflectes the network dynamics. For this the moving average equation is used as:

$$RT_{avg} = (1 - \gamma)RT_{avg} + \gamma RT_{val,cur} \tag{10}$$

where γ is the moving average co-efficient with value 0.1.

4.4 *Congestion detection for each node*

Based on the measured values of buffer level and RT_{val} each node can computes its congection level according to the condition of table 2. For buffer mesurment two buffer threshold Q_{max} and Q_{min} is used and the average buffer value is Q_{avg}. Condition 1 gives

ISBN: 978-960-474-361-2

us that if RT_{avg} is 0 or our buffer value is below the Q_{min} then there is no or low congestion. If the value of RT_{avg} fall between 0 and 1 or Q_{avg} is in between Q_{max} and Q_{min} then there is medium level congestion. The high level congestion detects when RT_{avg} grater than 1 or Q_{avg} is above Q_{max}. Two bits are used to indicate each congestion level.

Condition	Congestion level	Notification Bits
RT_{avg} =0 AND $Q_{avg} < Q_{min}$	No congestion	00
RT_{avg}=0 AND Q_{min}<Q_{avg}<Q_{max}	Low congestion	01
0< RT_{avg}<1 OR Q_{min}<Q_{avg}<Q_{max}	Medium congestion	10
RT_{avg}=1 OR Q_{avg}> Q_{max}	Highly congested	11

Table 2: Congestion Detection

4.5 *Rate Control*

The main goal of the proposed protocol is to detect congestion and try to avoid it. As the congestion level increases the number of packets drop increase. In our protocol always packet drops occur for condition 4 in table 2. The droppping packets selected from the reject state of buffer which has the lower priority packets compared to accept state. Also as node detects congestion it is necessary to control its service rate along with its child node. To control the rate the AIMD method is used here. Each node calculate the value δr. The δr is used to check whether a node will increase or decrease its packet service rate i.e R^i_{svc}.

Detecting the congestion level a node sends the congestion level value that is piggybacked in the forwarded data packets. Due to the broadcast nature of wirelesss network the message is heard by all neighnouring nodes. Hearing the message each node will

Congestion level	Value of δr	Rate of node i(R^i_{src})	Rate of Upstream node of i($R^{i,u}_{svc}$)
00	δr = 0	No Change	No Change
	δr>0	No Change	Increase
	δr<0	No Change	Decrease
01	δr = 0	No change	No change
	δr>0	No Change	No Change
	δr<0	Decrease	Decrease
10	δr = 0	No Change	Decrease
	δr>0	No Change	Decrease
	δr<0	Decrease	Decrease
11	δr = 0	Decrease	Decrease
	δr>0	Decrease	Decrease
	δr<0	Decrease	Decrease

Table 3: RateControl

Algorithm 1 Rate control at each node $i \in N$

1. Begin
2. **loop**
3. each i ∈ N calculate its congestion level and its own δ r(t). It then broadcast the congestion level message through its forwarded data packets
4. wait until it receives any congestion message or generates any congestion message
5. **if** (Table 2 returns no congestion or a node gets no congestion message from its parent) **then**
6. **if** (δr >0) **then**
7. $R^{i,u}_{svc}(t+1)$= $R^{i,u}_{svc}(t)$ + m, 0<m<1
8. **end if**
9. **if** (δr <0) **then**
10. $R^{i,u}_{svc}(t+1)$= $R^{i,u}_{svc}(t)$ × m
11. **end if**
12. **end if**
13. **if** (Table 2 returns low or a node gets congestion message low from its parent) **then**
14. **if** (δr <0) **then**
15. $R^i_{src}(t+1)$ = $R^i_{src}(t)$ × m
16. $R^{i,u}_{svc}(t+1)$ = $R^{i,u}_{svc}(t)$ × m
17. **end if**
18. **end if**
19. **if** (Table 2 returns medium or a node gets congestion message medium from its parent) **then**
20. **if** (δr <0) **then**
21. $R^i_{src}(t+1)$= $R^i_{src}(t)$ × m
22. **end if**
23. $R^{i,u}_{svc}(t+1)$= $R^{i,u}_{svc}(t)$ × m
24. **end if**
25. **if** (Table 2 returns high or a node gets congestion message high from its parent) **then**
26. $R^i_{src}(t+1)$= $R^i_{src}(t)$ × m
27. $R^{i,u}_{svc}(t+1)$= $R^{i,u}_{svc}(t)$ × m
28. **end if**
29. End
30. **end loop**

control its serving data rate according to value of δr. The idea of checking the value of δr is that if value of δr is less then 0 then the nodes may need not to increse the serving data rate. Depending the value of notification bit and the value of δr the source node and the upstream child node will control their rate as the table of Rate Control 3. The algorithm of rate control works as follows.

If a parent detects there is no congestion then it will not change its serving rate R^i_{svc}. It's child nodes i.e. the upstream nodes will increase their serving rate $R^{i,u}_{svc}$ when δr >0 (in line 7). δr >0 indicates that the outgoing rate is higher than the incoming rate of the parent. Since there is no congestion the children will increase their rate in a linear fashion that will increase the incoming rate of the parent. For the reverse case when δr <0 then child will decrease their rate to equalize the incoming and outgoing rate of the parent(line 10).

Getting congestion level low both the parent and upstream child will decrease their rate when δr <0. The child will decrease their $R^{i,u}_{svc}$ value and the parent will decrease its incoming rate R^i_{src} . This condition normally happens when current buffer is in the

ISBN: 978-960-474-361-2

filter state. As our protocol use congestion avoidance method so to avoid congestion both child and parent will regulate its outgoing flow (line 16,15).

The meduim congestion level happens when there is a possibility of link congestion or buffer overflow. For this case all children will decrease their outgoing rate $R^{i,u}_{svc}$(line 23). For link level congestion very soon the buffer of the parent will be full as packets can't be send successfully. So to remain safe all childdecrease their rate. The parent will decrease it's incoming rate R^{i}_{src} when it's incoming rate is larger than outgoing going rate (line 21).

When congestion level high all the nodes (parent and children) will decrease their rate R^{i}_{src},$R^{i,u}_{svc}$ rate as this is an alarming situation (line 26,27).

5 Performance Evaluation

In this section we evaluate the proposed protocol and compare its performance with other existing solutions. The topology used for the simulation is a tree based hierarchical static routing structure where sink is the root. The structure creates parent (downstream) and child (upstream) hierarchy among the nodes in the network. The sensed data could reach the sink with shortest number of hops. An event is generated at random location and in our simulation we have assign randomly the source IDs to the nodes within the event radius. We have added two additional fileds to modify

Area of sensor field	1000 1000 m^2
Number of sensor nodes	50
Radio range of a sensor node	70 m
Deployment type	Random
Buffer size	50
Packet length	64 bytes
Simulation time	250 seconds

Table 4: Simulation Parameters

the CSMA/CA MAC implementation for ns-3: notification bit and the Static Priority. The configuration of the simulation environment parameters are listed in table 4: The bursts of data traffic from four randomly chosen events, listed in table5 are considered in the performance studies. Three metrics, throughput, end-

	Event A	Event B	Event C	Event D
burst 1	20-40	30-60	25-75	35-85
burst 2	80-110	90-120	100-150	140-160

Table 5: Event and Brust Description

to-end delays, and weighted fairness, are selected to evaluate system performance. Throughput and delay are simulated and shown in fig 6 and fig 7 respectively. From fig 6 we can see that CFD has higher throughput than the other protocol as packets delivers here more fastly and shorter delay. The most impor-

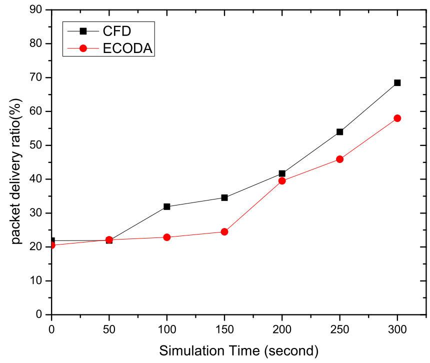


Figure 6: Throughput Comparison

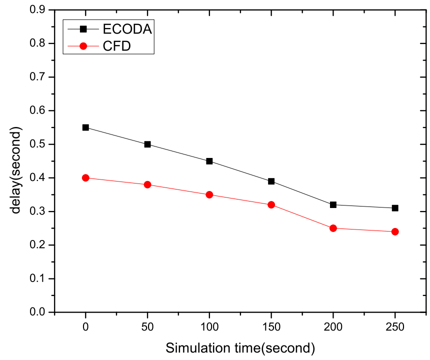


Figure 7: Delay Comparison

tant improvement of CFD is that it provides fairness to different class of traffic. The packet throughput and delay for different packets priorities are simulated and shown in fig 9 and fig 8. As, packets are scheduled according to their priority so the higher priority packets face low delay and better throughtput than other low priority packets.

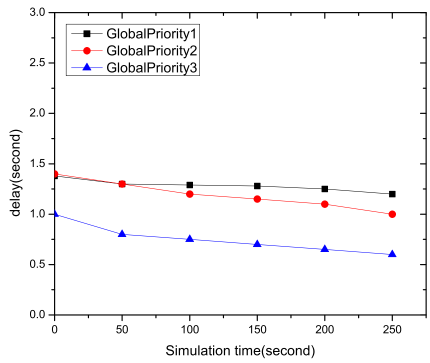


Figure 8: Delay for different Data packets

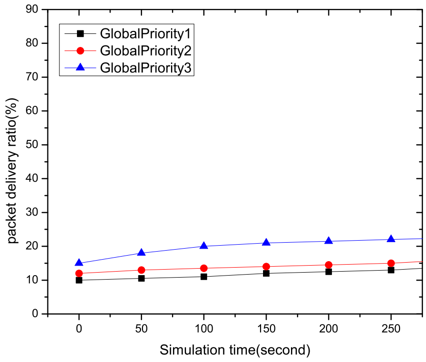


Figure 9: Throughput for Different Data Packets

6 Conclusion and Future Work

In this paper we propose a congestion control and rate adjustment protocol called CFD. CFD detects congestion level and it has a queue scheduler which ensures fair data delivery. CFD deals with transient congestion and persistent congestion efficiently. Through simulation it has been verified that CFD achieved high throughput and flexible fairness. It can reduce packet loss, improve energy efficiency, and lower delay. In future work, we will design the protocol for multipath. We are planning to extensively investigate this protocol performance including energy efficiency, buffer occupancy etc. through theoretical analysis.

References:

[1] C. Wan, S. B. Eisenman, and A. T. Campbell. Coda: congestion detection and avoidance in sensor networks. In *Proc. of the 1st international conference on Embedded networked sensor systems*, SenSys '03, pages 266–279, New York, NY, USA, 2003. ACM.

[2] J. Paek and R. Govindan. Rcrt: rate-controlled reliable transport for wireless sensor networks. In *Proc. of the 5th international conference on Embedded networked sensor systems*, SenSys '07, pages 305–319, New York, NY, USA, 2007. ACM.

[3] Özgür B. Akan and Ian F. Akyildiz. Event-to-sink reliable transport in wireless sensor networks. *IEEE/ACM Trans. Netw.*, 13(5):1003–1016, October 2005.

[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 *Proc. of the 5th International Conference on Embedded Networked Sensor Systems*, SenSys '07, pages 351–365, New York, NY, USA, 2007. ACM.

[5] S Rangwala, R. Gummadi, R. Govindan, and K. Psounis. Interference-aware fair rate control in wireless sensor networks. In *Proc. of the ACM SIGCOMM*, pages 63–74, 2006.

[6] L. Q. Tao and F. Q. Yu. Ecoda: enhanced congestion detection and avoidance for multiple class of traffic in sensor networks. *IEEE Trans. Consumer Electronics*, 56(3):1387–1394, 2010.

[7] C. Wang, M. Daneshmand, B. Li, and K. Sohraby. A survey of transport protocols for wireless sensor networks. *IEEE Network, May/June*, 20:34–40, 2006.

[8] F. Stann and J. Heidemann. RMST: reliable data transport in sensor networks. In *IEEE International Workshop on Sensor Network Protocols and Applications*, 2003.

[9] C. T. Ee and R. Bajcsy. Congestion control and fairness for many-to-one routing in sensor networks. In *Proc. of the 2nd international conference on Embedded networked sensor systems*, SenSys '04, pages 148–161, New York, NY, USA, 2004. ACM.

[10] C. Wang, B. Li, K. Sohraby, M. Daneshmand, and Y. Hu. Upstream congestion control in wireless sensor networks through cross-layer optimization. *IEEE J.Sel. A. Commun.*, 25(4):786–795, May 2007.

[11] M. M. Monowar, Md. O. Rahman, A. K. Pathan, and C. S. Hong. Congestion control protocol for wireless sensor networks handling prioritized heterogeneous traffic. In *Proc. of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, Mobiquitous '08, pages 17:1–17:8. ICST, 2008.

[12] K. Whitehouse, E. Sharp, C.and Brewer, and D. Culler. Hood: a neighborhood abstraction for sensor networks. In *Proc. of the 2nd international conference on Mobile systems, applications, and services*, MobiSys '04, pages 99–110, New York, NY, USA, 2004. ACM.

[13] M. A. Razzaque, C. S. Hong, and S. Lee. Autonomous traffic engineering for boosting application fidelity in wireless sensor networks. *IEICE Transactions*, 93-B(11):2990–3003, 2010.