Event Reporting with Fairness in Wireless Sensor Networks

FAISAL-BASHIR HUSSAIN AND YALCIN CEBI
Department of Computer Engineering
Dokuz Eylul University
Buca-Izmir, 35160 Turkey

Fairness in wireless sensor networks demands event nodes to have an equal share in the overall throughput of the system. Fairness is difficult to achieve in sensor networks due to multiple hop packet forwarding to a single destination that results in congestion. Moreover, greater the density of event region greater is the level of interference and greater is the number of packet drops. In this paper, we present a mechanism for fair event reporting that is implemented at the transport layer. The solution encompasses congestion control and fair rate adjustment schemes based on hop-by-hop packet delivery and buffer size. Moreover, a schedule based packet forwarding policy is used at the transport layer for ordered delivery of packets to the underlying layer; instead of commonly used jittered forwarding. The simulation results of our scheme shows high per node fair throughput from multiple event nodes to a single sink. Also, the use of our schedule-based packet forwarding policy helps to avoid packet collisions and increases the packet delivery ratio even from densely populated event regions.

Keywords: wireless sensor networks, event reporting, transport layer, fairness, congestion control

1. INTRODUCTION

Wireless sensor networks (WSNs) in recent years have emerged as key systems for information gathering. Sensors gather information from the environment through measuring mechanical, thermal, biological, chemical, optical, and magnetic phenomena. The electronics then process the information derived from the sensors and through some decision making capability direct critical event information to the sink.

Sensor nodes send sensing information either periodically or on event occurrence. An application may require general event region information, per node event information and time bound event information etc. [1, 2]. Consider a WSN working inside a mine for the purpose of environmental monitoring and control. The nodes monitor events like fire, oxygen content in air, sudden increase in pressure and leakage of toxic gases. For events like increase in temperature and pressure general event region information can be sufficient but for events like leakage of poisonous gas or oxygen content in air precise per node information is required to identify unsafe regions in the mine. Therefore, sensor to sink (many to one) event reporting in this regard can be categorized as follows:

1. Simple event reporting: Sensor nodes report event irrespective of per node throughput observed at the sink. As a result, nodes near the sink report more event packets than the nodes far away from sink. However, general event region information is obtained.
2. Fair event reporting: Sensor nodes adjust their reporting rates in order to fairly distribute the bandwidth among all the event reporting nodes. As a result per node reporting rate observed at the sink is same irrespective of node distance from the sink.

3. Prioritized event reporting: Sensor nodes use any metric like event, node, region or time as priority and adjust their reporting rates in order to provide prioritized event reporting.

In this paper, we address the issue of event reporting with fairness in wireless sensor networks while providing optimal throughput at the sink. Fairness can be achieved by assigning same reporting rate to all event reporting nodes and by avoiding congestion. Moreover, a packet forwarding scheme is required to ensure fair event packet delivery from a number of sources which can be at multiple hop distance from a single sink. Therefore, important issues concerning fairness are congestion control, fair rate adjustment and packet delivery.

Since a number of event sensing nodes try to send event information to the sink at the same time, sensors to sink flow causes congestion in WSNs [3, 6]. A node gets congested, if the incoming rate of packets is more than its outgoing rate which results in buffer overflow and packet loss. Another important reason of packet loss in sensor networks is interference. Greater the density of the network, greater will be the degree of interference; considering fix transmission power of nodes. Due to random deployment, the difference between densities in the same network will be even more extreme in “worst-case” networks [14]. In dense event regions contention for medium access increases, as packets do not get a chance for transmission in busy medium. This results into an increase in buffer size of nodes in dense networks. Hence, density of the network also adds to the degree of congestion in the network [10].

On event occurrence, the reporting rate of the event nodes is adjusted by the sink or by the nodes themselves, in order to obtain maximum event information. Sink based rate control generally assigns a single reporting rate to all the event reporting nodes within the network [5]. In case of congestion, the sink decreases the reporting rate of all the nodes, thus decreasing the overall throughput. While in case of node based rate control, each node increases or decreases its reporting rate depending on its local network conditions [6, 9]. But this result in unfairness, as the reporting rate of nodes far way from the sink are generally less than that of nodes near to the sink because of local congestion.

In this work, we consider flow based rate control in which the first hop node from the sink is responsible for adjusting the reporting rate of all the nodes routing through it. In flow based rate control, each first hop node and its associated flow is a separate entity. This entity is controlled by the first hop node, which assigns a schedule (reporting rate) to the nodes in the flow. According to congestion status of the flow, the first hop node can assign optimized reporting rate to the event nodes in the flow. Moreover, by assigning same reporting rate to all the nodes in the flow fairness can be achieved. Since each flow is a separate entity, the throughput of one flow can be higher than the other flow depending on local network conditions. Therefore, unlike sink based rate control, congestion in a single flow does not affect the reporting rate of the whole network.

In this paper, we propose a mechanism for fair event reporting that is implemented at the transport layer. It uses hop-by-hop packet delivery time and buffer size as the basic metrics for rate estimation and congestion detection. Moreover, we introduce a schedule-
based scheme at the transport layer for reporting rate assignment and ordered delivery of event packets to underlying routing layer. By using a schedule-based scheme at the transport layer helps to avoid packet collisions and increases the packet delivery ratio even in high densities. To the best of our knowledge, combined use of packet delivery time and buffer size for congestion control and the use of schedule based packet forwarding at the transport layer of wireless sensor networks had not been presented in any of the previous works.

The paper is organized as follows; next section presents the related work on fair event reporting and issues related to fairness like congestion control and rate adjustment. Section 3 presents the basic network setup and system definitions for the proposed fairness scheme. Section 4 presents the operation of the proposed of proposed scheme. In section 5 we show the simulation results and the last section concludes this paper.

2. RELATED WORK

In this section, we review several fair event reporting protocols along with congestion control and rate adjustment protocols for sensor networks.

Congestion Control and Fairness for many-to-one routing in sensor networks (CCF) [4] uses buffer size to detect congestion. CCF implements a tree based technique in which each node calculates its sub-tree size. Reporting rate is allocated to nodes depending on their sub-tree sizes. Every node maintains a separate queue for each of their previous hop nodes. Nodes forward packet from these queues depending on the sub-tree size of the previous hop nodes during each epoch. Sensor nodes have limited memory resources, maintaining a separate queue for each previous hop node is not a memory efficient solution; especially in dense networks. Also, in case of multiple events CCF treats all events similarly which can have different reporting rate requirements.

Credit based fairness control in wireless sensor network (CFRC) [15] allocates bandwidth to nodes based on effective amount of sensed information which is dependent on node density and their distribution instead of uniformity. However, CFRC does not provide per node fairness instead provides fairness on credit basis depending on sensed information by a node. Mitigating congestion in wireless sensor networks [12], suggests multiple approaches for congestion removal which span on different layers of the traditional protocol stack. However, these schemes are applicable for networks in which nodes offer same traffic load and the routing tree is not significantly skewed. Priority-based congestion control protocol (PCCP) [8] uses packet inter arrival time and packet service time to detect congestion level at a node and employs weighted fairness to allow nodes to receive priority-dependent throughput. However, PCCP focuses on priority dependent throughput not on fairness.

Event-to-Sink reliable transport in wireless sensor networks (ESRT) [5], uses change in local buffer level of a node during consecutive intervals to predict for congestion. In case of congestion the sink regulates all the sources in the network regardless of a particular nodes causing congestion; this decreases the overall system throughput. Apart from this ESRT does not implement any mechanism for fair packet delivery from nodes to sink. A price-oriented reliable transport protocol (PORT) [7], uses link loss estimation as a basic source of congestion detection and avoids congestion by dynamically forward-
ing packets to less congested nodes. In PORT, sink directs individual nodes to increase or
decrease their reporting rates. However in dense networks, sending such control information
to every node is very difficult, since nodes can be at multiple hop distance from the
sink. Interference-Aware fair rate control in wireless sensor networks (IFRC) [11],
detects congestion by monitoring average queue length and exchanges congestion state
among the potential interferers using a congestion sharing mechanism. IFRC only takes
effect after congestion happens, it cannot mitigate congestion and avoid packet drops.

Congestion detection and avoidance in sensor networks (CODA) [3], uses open loop
and closed loop mechanisms to handle congestion which is detected on the basis of channel
sampling and buffer occupancy. SenTCP [6], uses hop-by-hop, open loop congestion control mechanism that detects congestion using both buffer occupancy and packet inter-arrival time. In both CODA [3] and SenTCP [6] nodes depending on their local network conditions adjust the reporting rate of their previous hop node irrespective of per node throughput at sink, hence fairness can not be achieved using these protocols.

In short, congestion in sensor networks results into packet drops and unfair throughput at the destination. The existing fairness protocols use metrics like buffer size, packet service time, packet inter-arrival time and load assessment, to detect congestion. However, in variable node densities these metrics when used alone are unable to detect congestion properly. Moreover, congestion is not the only source of packet drops in sensor networks. At high node densities, packet drops can occur due to interference. Therefore, this work presents not only a novel congestion detection mechanism based on both buffer size and hop-by-hop packet delivery time, but also a schedule based packet forwarding scheme to reduce packet drops due to interference.

3. SYSTEM DEFINITION

In this section, we explain the basic system related definitions, network setup and
assumptions. The network comprises of non mobile wireless sensor nodes and a sink. We
define the nodes as event reporting, routing, reporting & routing, and idle nodes. If $b, c, d$
are the nodes routing through node $a$ then $b, c, d$ are the previous hop or child nodes of $a$
and $a$ is the next hop node of $b, c, d$ as shown in Fig. 1. All nodes routing event informa-
tion through node $a$ are associated with same information flow.

Fairness can be categorized as either node or event based fairness. In node based
fairness, the sink receives same number of packets from all the nodes, irrespective of
node distance from the sink. In order to assure fairness, each node is assigned reporting rate according to their sub-tree sizes (the number of nodes traversing through a node) and not according to the initial reporting rate of event(s) that are being reported. Therefore, multiple events with different initial reporting rates can not be handled. For example, in Fig. 1, if nodes $b$ and $d$ are reporting two different events having different initial reporting rates, then node $a$ can not assign different reporting rates to these nodes, as their sub-tree size is same. On the other hand, in event based fairness the sink receives same number of event packets from all the nodes reporting same event, irrespective of node distance from the sink. If all event nodes are reporting the same event, then all event nodes will get equal share of the link bandwidth resulting into fairness; similar to node based fairness. However, in case of multiple events, nodes according to their initial event reporting rates
will get share of the link bandwidth. We define event reporting to be fair if the sink receives same number of event packets from all event nodes reporting the same event.

In order to achieve minimum hop routing, each node maintains a next hop table which contains the list of its possible next hop nodes, which are at minimum hop distance from the sink. The table is established during the initial network setup as the sink broadcasts a route discovery packet. The route discovery packet includes source ID (sink), sender ID and hop count. Neighboring nodes receiving the packet add the sending node into their next hop table and increments the hop count of the packet. A node receiving a route discovery packet only forwards it to the next hop node if the hop distance of the received packet is smaller than or equal to the already stored hop distance, otherwise it discards the packet. Hence, next hop table is established at network setup time using controlled flooding. Nodes during event reporting randomly select a single next hop node from their table and forward their packets through that node.

Nodes maintain successive fixed size data ($\gamma$) and schedule ($\delta$) intervals throughout their life time. During the data intervals nodes generate/route available event information and during schedule intervals routing nodes send transmission schedule for their previous hop nodes. The schedule comprises of slot length ($\lambda$ sec), total number of slots and allocated number of slots for a previous hop node. We define slot in terms of a time duration during which a node can forward a single packet.

Let us assume for simplicity that all nodes except node $a$ in Fig. 1 are event reporting nodes. Node $a$ will generate a schedule for its previous hop nodes which is shown in Table 1, (details of slot length calculation and slot allocation are explained in the next section). Nodes $b$, $c$ and $d$ will divide their data interval into 0.1 second intervals, which is the initial slot length on event occurrence. These nodes will forward one event packet to node $a$ during their allocated slots. Further more, node $c$ will generate and forward a schedule for node $e$ and $f$ as shown in Table 2.

If there is no interference between two nodes, then slot reuse is possible. Slot reuse can increase the overall throughput of the system but will result in unfair per node throughput at the destination. For example, if $d$ and $f$ are disjoint nodes in Fig. 1 and they use same slots, then total number of slots will be five. This will allow node $a$ to send five packets to the sink per single data interval from six child nodes, resulting into unfair throughput. Assuming another implementation of slot reuse, if the total numbers of slots are kept six and nodes $d$ and $f$ reuse slot five, then this can increase the throughput on the cost of a more complicated scheduling scheme. Also in this case, each node will have to additionally monitor the interfering and non-interfering neighboring nodes. Therefore, slot reuse is not considered in the proposed scheduling scheme.

Fig. 1. Flow of event and schedule packets.
Table 1. Transmission schedule generated by node $a$.

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Total Slots</th>
<th>Initial Slot</th>
<th>End Slot</th>
<th>Slot Length (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>b</td>
<td>6</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>c</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>0.1</td>
</tr>
<tr>
<td>d</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Table 2. Transmission schedule generated by node $c$.

<table>
<thead>
<tr>
<th>Node ID</th>
<th>Total Slots</th>
<th>Initial Slot</th>
<th>End Slot</th>
<th>Slot Length (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>e</td>
<td>6</td>
<td>3</td>
<td>4</td>
<td>0.1</td>
</tr>
<tr>
<td>f</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>0.1</td>
</tr>
</tbody>
</table>

Slot length determines the reporting rate during an interval. If the slot length is short, more traffic will be forwarded by the node and vice versa. TDMA based techniques use fixed time slots that are assigned to sources, which send information during their specified time slots. However, in our schedule-based scheme time slots are dynamically assigned during the schedule intervals depending on the average packet delivery time observed by node and buffer size. We define packet delivery time as the time a packet takes to reach from the transport buffer of a previous hop node to the next hop node’s transport buffer. Nodes maintain a queue at the transport layer and forward a single packet from it during their allocated slots. Packet delivery time not only includes service time but also the transmission time plus the reception time at the destination.

We divide the buffer size of nodes into three ranges low, medium, and high. Proposed fairness scheme controls congestion by maintaining the buffer size of nodes within the medium range. The medium range in this work is referred as the optimal range. Since buffer size increases relatively slowly as compared to the actual load on the channel [3, 6], using high range as optimal can result in buffer overflows. If the buffer size is low/high nodes will decrease/increase the slot length for the next interval respectively to achieve optimal range; otherwise nodes will maintain the slot length for the next interval. Hence, each node adjusts the reporting rate (through slot length) of its previous hop nodes, so that the buffer is optimally utilized and to avoid congestion while providing high throughput.

4. THE PROPOSED FAIRNESS SCHEME

In this section the proposed fairness scheme is presented that comprises of fair slot allocation, slot length calculation and the general operation of the scheme.

Fair slot allocation allows nodes to get slots for transmission during a data interval with respect to the traffic that is traversing through the node. Slot length calculation mechanism uses packet delivery time and buffer size of nodes to calculate an appropriate slot length in order to provide maximum throughput while avoiding congestion. The operation of this scheme comprises of consecutive data and schedule intervals that ensure fair event reporting while minimizing the element of interference. Therefore, in the remainder of this section we explain fair slots allocation, slot length calculation, and the operation of packet scheduling scheme.
4.1 Fair Slot Allocation

A slot is a time duration during which a node can forward a single packet. Therefore greater the number of slots assigned to nodes greater will be their reporting rate. In order to achieve fair event reporting, nodes are assigned slots equal to their initial event reporting rate.

Nodes report event by indicating event’s initial reporting rate (application defined for a particular event) to one of its next hop node; during data interval. The next hop node forwards its updated total reporting rate which is the sum of its own event reporting rate (in case of event node) and reporting rate of all the nodes traversing through it. In this way, the first hop node from the sink obtains the total reporting rate of the flow. The first hop node after receiving the initial event request assigns slots to their previous hop nodes during schedule interval. Each previous hop node is assigned slots according to its total reporting rate. The previous hop nodes further divide the received schedule among their child nodes; according to their total reporting rates. If a node is both event reporting & routing node, then the schedule contains slots for itself, as well as for its previous hop nodes, according to their reporting rates.

If event $E_1$ has an application defined initial reporting rate $R_1$ which is the minimum reporting rate ($R_{min}$) among all the events observed by the network. In order to simplify slot assignment, we assume that reporting rates of other events will be a factor of $R_{min}$, expressed as $2^n R_{min}$ while $n \geq 0$. Let $N_i$ be the total number of previous hop nodes routing through a first hop node $i$. The number of slots $S_j$ assigned to the $j$th previous hop neighbor of the $i$th node can be calculated, based on the total reporting rate $R_j$ of the $j$th node as: $R_j / R_{min}$; where $R_j \in 2^n R_{min}$ and $R_j \geq R_{min}$. The total number of slots allocated to all the previous hop nodes of $i$ will be $\sum_{j=0}^{k} R_j / R_{min}$. For example, in Fig. 1 all nodes are event reporting nodes reporting same event, expect first hop node $a$ which is a simple routing node only. The initial reporting rate of all event nodes is same. Let initial reporting rate be 10 packets per interval. Then the total reporting rate of node $b$, $c$, $d$ will be 10, 40, 10 respectively, while node $a$ has a total reporting rate of 60. Therefore, node $a$ will assign slots to previous hop nodes according to their total reporting rates as shown in Table 1. Node $c$ has further child nodes therefore it will divide its slots among its child nodes according to their total reporting rates, as shown in Table 2. Since node $c$ itself is an event reporting node therefore it allocates slots for its self too.

Since the event region can expand after event occurrence, new nodes can join event reporting. Also fresh nodes can be scattered or thrown in the event region. As new nodes join the network, they will forward their initial reporting rates, as they detect the event. The initial reporting rates of these nodes will be forwarded to the first hop node. The first hop node and all the intermediate nodes will update their total reporting rates and total number of slots. Thus, new nodes will be issued slots during the next schedule interval to report the event.

Nodes can stop reporting event due to battery failure, removal of the detected event or because of a physical damage. In case of battery failure and removal of the detected event, nodes will inform their next hop nodes about this situation, prior to discontinuing event reporting. Likewise, their next hop nodes will forward this information to the first hop node. The first hop node and the intermediate nodes will delete the node from their schedule and will decrease their total reporting rates. In case of physical node failures, it
is not possible to forward the discontinuing information to the next hop nodes. Therefore, each node can monitor the slot usage during a data interval. If nodes observe that any of their previous hop nodes has not sent a single data packet during the data interval, then they can consider the node as dead. Furthermore, this information can be forwarded to the first hop node.

4.2 Slot Length Calculation

In order to provide optimal throughput while avoiding congestion the slot length is calculated by nodes depending on local network condition; in terms of average packet delivery time and buffer size.

Nodes observe the average packet delivery time of their previous hop nodes in a data interval from the received packets. The average packet delivery time observed during the data interval is used as the slot length for the next data interval. However, if a node’s buffer is either under or over utilized during the data interval then the node adjusts its slot length in order to optimally utilize the buffer.

Algorithm 1, illustrates the slot length calculation procedure. Let $\lambda_i^t$ be the slot length for the $t$th interval; $B_i^t$ and $B_i^{t-1}$ be the buffer size for the $t$th and $(t-1)$th interval of $i$th node. Then, in order to calculate appropriate slot length for the next interval $\lambda_i^{t+1}$, a node measures change in buffer occupancy ($\phi_B$) and the predicted buffer occupancy ($\rho_B$) for the next interval; similar to ESRT [5] as:

$$\phi_B^{t+1} = B_i^t - B_i^{t-1} \quad \text{and} \quad \rho_B^{t+1} = B_i^t + \phi_B^{t+1}.$$  \hfill (1)

In case the predicted buffer occupancy is not in the optimal buffer occupancy ($O_p B_{min} \leftrightarrow O_p B_{max}$) range, then nodes adjust their slot length for the next interval by adding or subtracting a deviation factor ($\omega$) in the current slot length. We calculate deviation factor $\omega^t$ for the $i$th node, at the end of $t$th interval as:

$$O_p B = \frac{(O_p B_{min} + O_p B_{max})}{2} \quad \text{(2)}$$

Deviation ($\omega^t$) = $\pm \frac{(O_p B - \rho_B^{t+1})}{O_p B} \ast \lambda_i^t$. \hfill (3)

Hence, the slot length for the $(t+1)$th interval will be:

$$\lambda_i^{t+1} = \lambda_i^t - \omega^t \quad \text{(if} \ \rho_B^{t+1} < O_p B_{min})$$

$$\lambda_i^{t+1} = \lambda_i^t + \omega^t \quad \text{(if} \ \rho_B^{t+1} > O_p B_{max})$$

$$\lambda_i^{t+1} = \lambda_i^t \quad \text{(if} \ \rho_B^{t+1} \in O_p B_{min} \leftrightarrow O_p B_{max}).$$

A zero buffer occupancy during an interval means that the link is under utilized and the packet forwarding rate of previous hop node is low. Therefore, if the buffer occupancy is zero the slot length for the next interval is decreased to half.
Algorithm 1  SLOT LENGTH CALCULATION: This procedure is called at the start of each schedule interval by routing and reporting & routing nodes.

1. CURRENT_SLOT_LENGTH ($\lambda_t$) = AVERAGE_DELIVERY_DELAY
2. if BUFF_SIZE = AVERAGE_DELIVERY_DELAY then
   3. /* Initially when event is detected and reported */
   4. NEXT_SLOT_LENGTH ($\lambda_{t+1}$) = 0.1 sec (Default)
   5. GO TO STEP 22
6. end if
7. if BUFF_SIZE = 0 then
   8. /* Special case when reporting rate is low */
   9. $\lambda_{t+1} = \lambda_t / 2$
10. GO TO STEP 22
11. end if
12. Calculate PREDICTED_BUFF_SIZE ($\rho_B$) * Using Eq. (1) */
13. if ($\rho_B < O_p B_{min}$) then
   14. Calculate DEVIATION FACTOR ($\omega$) * Using Eq. (3) */
   15. $\lambda_{t+1} = \lambda_t - \omega$
16. else if ($\rho_B > O_p B_{max}$) then
   17. Calculate DEVIATION FACTOR ($\omega$) * Using Eq. (3) */
   18. $\lambda_{t+1} = \lambda_t + \omega$
19. else
20. $\lambda_{t+1} = \lambda_t$
21. end if
22. TRANSMIT SCHEDULE

4.3 The Operation

In this section we present, the operation of our fair event reporting scheme that is controlled by the first hop node from the sink. Since, each node maintains a data and schedule interval therefore the operation of the fairness scheme can be subdivided into data interval operation and schedule interval operation.

4.3.1 Schedule interval

The slot length received from a parent node or next hop node is termed as basic slot length (BSL) while the slot length calculated by the node itself is termed as local slot length (LSL). For first hop nodes LSL and BSL are same. The first hop nodes send their schedule packets to their previous hop nodes. Each previous hop node compares the received BSL with their LSL. There are three possibilities local slot length less than, greater than or equal to basic slot length.

1. $LSL < BSL$: If a node has local slot length less than basic slot length for the current interval this means the node will generate more traffic than requested from the next hop node. This will result into unfairness as well as congestion at the next hop node. Therefore, the local node accepts the basic slot length and passes it to their child nodes.
2. $LSL > BSL$: If a node has local slot length greater than basic slot length this means the node will generate less traffic than requested from the next hop node. This situation
occurs if the node receiving the schedule is locally more congested than its next hop node. In this case, LSL is forwarded in the schedule packets to child nodes. Although this results into temporary unfairness, as the node forwards less traffic than requested from next hop node but helps to mitigate local congestion.

3. $\text{LSL} \approx \text{BSL}$: In this local and basic slot lengths are approximately equal therefore the nodes will send basic slot length to their child nodes.

### 4.3.2 Data interval

Each node maintains intervals equal to their selected slot length during the data intervals. At the expiry of each slot interval a node checks whether it is allowed to send packet during the interval according to its schedule. Event reporting nodes during the allocated slots sends an event packet to the next hop node. Routing nodes simply forward a packet from the transport buffer to its next hop node in its allocated slots. However, event reporting & routing nodes send new event packets as well as forward event packets from the transport queue according to their schedule.

Schedule packets are a natural overhead in the proposed fairness scheme. The numbers of schedule packets in our fairness scheme are very less and constant, as compare to data packets. Let, $K$ be the number of event reporting nodes while $N$ be the number of non-idle nodes in a single flow. In this case, $N - 1$ will be the number of schedule packets generated during each schedule interval.

If $\lambda$ is the slot length during a data interval set by the first hop node of the flow, then $(1/\lambda) \times \gamma$ will be the number of data packets delivered to sink by the flow. $\lambda$ is adjusted by nodes and its value is basically dependent on average packet delivery time, observed during a data interval. Therefore, the value will be sufficient to deliver a single packet within the slot length. Length of data interval ($\gamma$) is fixed. Longer the length of data interval, smaller will be the overhead of schedule packets. On the other hand, smaller the length of data interval greater will be the overhead of schedule packets. Also, random will be the behavior of fairness scheme, as the reporting rate is adjusted frequently. Therefore, length of data (4 sec) and schedule intervals (1 sec) are mirrored to provide high throughput with low overhead.

### 5. SIMULATION RESULTS

We show the performance of our Proposed Fairness Scheme (PFS) using network simulator ns2 [13], which is a scalable discrete-event simulator. We evaluate the performance of proposed fairness scheme in terms of fair event reporting at sink, packet receive ratio, multiple events and energy consumption.

The proposed fairness scheme uses minimum hop packet forwarding at the routing layer. Length of data interval is 4 second while schedule interval is 1 second. The sink observes 10 second intervals to evaluate the output of the fairness scheme. The length of sink interval can vary, according to the nature of application or events observed. However, events generally occur for small duration of times in sensor networks [5]. Assuming the mining application, 10 seconds is a sufficient time to detect leakage of a poisonous gas,
which should be evaluated quickly and fairly. Table 3, summarizes the configuration parameters for the simulations.

We compare our simulation results with CCF [4]; commonly referenced fairness scheme for wireless sensor networks. In CCF, nodes use packet service time to predict for their reporting rates. Furthermore, nodes use buffer size to predict for congestion. Each node implements a separate child node queue at the transport layer. Fairness is achieved by forwarding packets from the child node queues equal to sub-tree size of each child node.

Table 3. Simulation parameters.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport Layer</td>
<td>Proposed Fairness Scheme</td>
<td>802.11</td>
</tr>
<tr>
<td>Network Layer</td>
<td>Minimum Hop Forwarding</td>
<td>36 bytes</td>
</tr>
<tr>
<td>MAC Layer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Packet Length</td>
<td></td>
<td>65 Packets</td>
</tr>
<tr>
<td>IFQ Length</td>
<td></td>
<td>50 Packets</td>
</tr>
<tr>
<td>Transport Queue</td>
<td></td>
<td>0.660 W</td>
</tr>
<tr>
<td>Transmit Power</td>
<td></td>
<td>0.395 W</td>
</tr>
<tr>
<td>Receive Power</td>
<td></td>
<td>20m</td>
</tr>
<tr>
<td>Radio Range</td>
<td></td>
<td>4 sec</td>
</tr>
<tr>
<td>Data Interval</td>
<td></td>
<td>1 sec</td>
</tr>
<tr>
<td>Schedule Interval</td>
<td></td>
<td>10 sec</td>
</tr>
<tr>
<td>Sink Interval</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.1 Fair Event Reporting

Event reporting is fair if the per node throughput at the sink is same for all event reporting nodes in the flow. Fig. 3, shows the simulation scenario where 20 nodes are reporting the same event through first hop node “0” to the sink. In order to decrease multiple hop interference, the radio range used in this simulation is 10 meters. The arrow heads show the direction of minimum hop routing. The sink is located at (90, 50) coordinates in a 100 × 100m sensor field with maximum hop distance of 9 hops.

Fig. 3, shows the per node throughput observed at the sink during 100 sec of event reporting. We compare the per node throughput of our scheme with CCF [4] and Simple Event Reporting (SER). In simple event reporting nodes use Additive Increase Multiplicative Decrease (AIMD) to adjust their reporting rates and simple buffer occupancy is used for congestion control. In case of simple event reporting, we use an initial reporting rate of 2 packets per second while the reporting rate is increased by a factor of 1.2 during each interval and decreased to half in case of congestion. Simple event reporting does not use any explicit fairness mechanism therefore per node throughput at sink is variable. CCF [4] provides considerably fair output but Proposed Fairness Scheme (PFS) provides even better results with high per node throughput. Since each node is assigned a schedule, in order to guarantee fair per node throughput.

We further observe the per node throughput of our fairness scheme under single hop interference and multiple hop interference.
5.1.1 Single hop interference

For evaluating the performance of proposed fairness scheme in single hop interference, event nodes are placed in an event region centered at (60, 60) coordinates. The diameter of event region is 10 meters. All nodes are arranged so that they are at a single hop distance from the first hop node. The sink is at (90, 90) coordinates. Figs. 4 (a) and (b), show the per node throughput of 50 and 100 event reporting nodes using PFS and CCF. PFS provides high per node throughput than CCF in both node densities. The performance of CCF severely degrades in high density (100 event nodes) because of increased packet drops due to interference.

5.1.2 Multiple hops interference

In this case, the event nodes are randomly placed in an event region centered at (40, 40) coordinates. The event region has a diameter of 40 meters and the sink is at (90, 90) coordinates. 50 and 100 event nodes report event to the sink through a single first hop node. From Figs. 5 (a) and (b), it is evident that PFS in both node densities provide more fair and high per node throughput than CCF.
5.2 Packet Receive Ratio

Packets reception ratio decreases either by an increase in the reporting rate or by an increase in the density of the network (considering fixed transmission power for nodes). Since both congestion and interference results in packet drops. The simulation scenario discussed in section 5.1.2 is used to evaluate the packet receive ratio. Since the density of event reporting nodes is less, CCF provides high packet receive ratio as shown in Fig. 6 for 50 nodes. However, due to interference and busy medium at high density CCF fails to provide high packet receive ratio for 100 nodes. The packet receive ratio of our fairness scheme is above 95% under low as well as high densities. This is because of our congestion control mechanism that not only avoids but controls congestion by efficiently adjusting the reporting rate of congested nodes, resulting in few packet drops. Moreover, schedule based packet forwarding provides an upper layer solution for packet drops due to interference.
5.3 Scheduled vs Jittered Forwarding

Fig. 7, compares our schedule-based scheme with a simple jittered based forwarding scheme when both are applied at transport layer. In both these schemes, packet delivery time and buffer size are used to predict for congestion. In jittered forwarding version routing nodes send reporting rate to their previous hop node instead of a schedule. In order to achieve fair per node throughput like CCF, each node forwards packets equal to its sub-tree size. Fig. 7, shows that per node throughput observed at the sink for 100 randomly distributed event reporting nodes in an event region centered at (40, 40) coordinates with an event diameter of 40m. We have conducted simulations for 120 sec of event reporting. In case of jittered forwarding due to high density, collisions and busy medium; the buffer of nodes start to overflow resulting into packet drops. However, by using scheduled transmissions packet delivery ratio increases resulting into fair and increased throughput.

5.4 Multiple Events

In wireless sensor networks more than one event can occur at the same time. Fig. 8 shows the per node throughput of two different events. In the simulation scenario, nodes are arranged as shown in Fig. 2. Each of nodes 3, 9, 11, 17 and 19 start reporting an event at 4 packets per second while the rest of the nodes report a different event at 2 packets per second. Since CCF only considers node based fairness, it is unable to provide fair event reporting with respect to multiple event demands. Proposed fairness scheme uses initial event reporting rate for rate allocation, therefore nodes according to the event’s demand get a share of the bandwidth. Although PFS shows fair per node throughput at the sink but the overall throughput of PFS (74 packets per second) is slightly less than CCF (77 packets per second). The natural intuition behind this slight decrease is that the first hop node, in order to assure fairness does not increase the overall throughput on the cost of unfair per node throughput.

![Fig. 8. Per node throughput of 20 event nodes reporting two different events.](image)

![Fig. 9. Residual energy of 100 and 150 event nodes during 150 sec of event reporting.](image)
5.5 Energy Consumption

In Fig. 9, we show the total energy utilization of 100 and 150 event reporting nodes in 100 × 100m sensor field. Initial energy of all the nodes is set to 0.1 Joules for this simulation. Despite of additional scheduled packet transmission, proposed fairness scheme decreases the energy consumption since it handles congestions efficiently and also because it does not increase the reporting rate of nodes more than they can handle.

6. CONCLUSION

A fair event reporting scheme for wireless sensor networks has been presented in this paper. Simulation results have confirmed that the rate adjustment mechanism, efficiently and equally distributes the maximum achievable throughput among all the event reporting nodes. Also, congestion control based on hop-by-hop packet delivery time and buffer size reduces packet drops due to congestion. The packet drops due to interference at high densities are reduced by the use of our scheduling mechanism at the transport layer. As a result, the overall throughput and packet receive ratio increases while a decrease in energy utilization of nodes is observed.

REFERENCES

9. S. Chen and N. Yang, “Congestion avoidance based on light-weight buffer management in sensor networks,” IEEE Transactions on Parallel and Distributed Sys-
Faisal-Bashir Hussain and Yalcin Cebi


Faisal-Bashir Hussain received the B.Sc. degree from Punjab University, Lahore, Pakistan in 1999. He got the M.S. degree in Computer Science from International Islamic University, Islamabad, Pakistan in 2002. He is currently pursuing his doctoral degree in Computer Engineering from Department of Computer Engineering, Dokuz Eylul University, Izmir, Turkey. His current research area includes wireless network, ad-hoc network, and wireless sensor network.

Yalcin Cebi received the B.S. degree from Dokuz Eylul University, Izmir, Turkey in 1986. He got the M.S. and Ph.D. degrees in Mining Engineering at Dokuz Eylul University from Graduate School of Natural and Applied Sciences, Izmir, Turkey, in 1988 and 1995, respectively. He is currently an Associate Professor in the Department of Computer Engineering, Dokuz Eylul University, Izmir, Turkey. His current research area includes computer networks, natural language processing, wireless networks, ad-hoc networks, production planning and optimization.