# ClearBurst: Burst Scheduling for Contention-Free Transmissions in Sensor Networks

Ren-Shiou Liu, Kai-Wei Fan, and Prasun Sinha
Department of Computer Science and Engineering
The Ohio State University
Email: {rsliu, fank, prasun}@cse.ohio-state.edu

***Abstract*— In wireless sensor networks, the many-to-one data communication pattern induces high collision losses as multiple transmissions cause contention and interference along the paths from sources to the sink. This paper proposes a low-overhead MAC layer solution to address the high contention problem to improve system throughput and reduce energy consumption. Periods of bursty transmissions with reduced contention from neighboring nodes are exploited to efficiently clear up backlogged queues and improve the performance of CSMA. Through analytical modeling we characterize the expected performance improvement and show that it conforms to the simulation results. Using extensive simulations on *ns-2* and experiments on the $49$ node sensor network testbed (*Kansei*) running TinyOS we evaluate the performance of our solution.**

## I. INTRODUCTION

Energy and channel capacity are two critical resources in wireless sensor networks. When a large number of nodes start reporting data, sensor networks easily get overwhelmed by high contention and interference along adjacent multihop routing paths and in the neighborhood of data collection points such as the sink. This leads to inefficient use of these resources. Various approaches have been proposed to mitigate this problem, such as improved MAC layer designs [1], [2] and back-pressure techniques at the link layer [3]–[6]. In [1], a hybrid TDMA/CSMA approach is proposed to address congestion near the sink. However, it requires specific capabilities only available at the sink. ZMAC [2] is another hybrid TDMA/CSMA based solution, but it requires time synchronization and distributed slot assignment using the DRAND [7] protocol, which significantly increases the complexity and overhead of the protocol. In addition, computation of the TDMA schedules is expensive in dynamic environments where the traffic sources change with time. Back-pressure based mechanisms for congestion control [3]–[6] operate over the MAC layer to maintain the queue size at acceptable levels to avoid queue drops. As these mechanisms are not integrated into the MAC layer where congestion is first observed, their impact on performance improvement is limited.

*In this paper we seek to design a low-overhead MAC layer solution to address the overload problem in wireless sensor networks*. Our solution is based on the observation that throttling sensors' reports to prevent simultaneous transmissions can reduce contention and increase throughput. We propose a burst scheduling approach at the MAC layer specifically designed to mitigate the overload problem. The scheduling overhead is reduced as a burst of packets as opposed to a single packet is scheduled for transmission. If a node observes an increase in its queue backlog, it performs a low-overhead coordination with neighboring nodes to reserve a period for transmitting a packet burst. By alleviating contention during the burst periods, throughput is boosted for transmissions from sources to the sink. In addition, by explicitly addressing backlogged queues, overall queue drop rate decreases and network performance is improved.

We make the following contributions in this paper.

- We propose ClearBurst for mitigating network overload which does not incur the overheads of TDMA based approaches. Moreover, it is applicable at any node *anywhere* in the network.
- We present results from experimentation on a large-scale indoor testbed based on implementation on TinyOS.
- We perform extensive evaluation using *ns2*.
- We analytically model the expected performance gains for representative network scenarios by extending the analysis techniques used in Bianchi's work [8], [9].

The organization of the rest of the paper is as follows. Section II and III present our proposed approach and analytical modeling of the proposed solution. Simulations and experimental results are presented in Sections IV and V. Section VI concludes the paper.

## II. DESIGN OF CLEARBURST

To mitigate the contention and overload problems, we propose *ClearBurst* in the MAC layer to coordinate media access control for sensor nodes. ClearBurst uses dedicated slots for burst transmissions to and from an elected node called a C-node. The burst transmission reduces contention and interference resulting in reduced energy consumption and increase in overall throughput.

**C-node Election** In ClearBurst, first a C-node is elected for a set of sources to act as a data collection point as well as a schedule coordinator. Although TDMA-based approach can reduce the contention, it incurs high overhead for time synchronization and slot assignment. Using a C-node as the coordinator not only eliminates these overhead but also makes the schedule adaptive to dynamic traffic and unpredictable topological changes. In addition, C-node can serve as an aggregation point which aggregates raw data packets and reduces the amount of information transmitted in the network.

To reduce the time slots reserved for burst transmissions, a node closest to the sink among all source nodes is first elected as the C-node, and a tree rooted at the C-node containing all source nodes is created. Many cluster-head election and tree construction algorithms have been proposed and can be adopted. For example, the cluster-head election, tree construction and migration approach described in [10] can be used to elect the C-node and to construct and maintain the tree. After the C-node is elected and a tree is constructed, source nodes send their packets to the C-node along the tree, and the C-node forwards these packets to the sink.

**Congestion Detection** When the traffic load is low, nodes use a CSMA-based protocol to transmit their packets since CSMA-based approaches perform well in low traffic scenario. However, when the traffic load is high, its performance drops significantly due to congestion caused by high contention and collision. Various congestion indicators have been proposed, such as monitoring the number of packets in the queue [3], [5], [6], sampling the channel periodically [4], exchanging queue length information with neighbor nodes [11]. To minimize the overhead, we adopt queue occupancy as the congestion indicator in ClearBurst. When the number of packets in the transmission queue exceeds a predefined threshold, ClearBurst steps in and starts coordinating the transmissions. As the bottleneck is likely to happen around the C-node, ClearBurst coordinates the transmissions only for nodes near the C-node to minimize the control overhead. Multiple C-nodes for coordination is possible if these C-nodes are not interfering with each other and we leave it as the future work.

**Burst Scheduling** To start the coordination, child nodes of the C-node signal the need for burst transmission by setting the request in the data packet header. When the C-node receives a packet with the request and if it is not serving any burst transmission, it grants the request by piggybacking the acknowledgement in outgoing data packets. The child node can overhear the data packets and know its request has been granted, and we call the child node an active node. The request and acknowledgement handshake serves the purpose of reserving the channel for burst transmission. Because the interference range are usually larger than communication range, this scheduling information needs to be propagated to nodes that may interfere with the burst transmission.

To make sure that these potential contending nodes and interfering nodes are shut off during burst periods, ClearBurst uses a small time window after the handshake to propagate the scheduling information before starting the burst period. During the small time window, nodes can still access the channel using CSMA, but all the nodes who have learned the schedule information by overhearing propagate the information by piggybacking it in every outgoing data packet. The number of hops to propagate the information can be controlled by TTL filed in the header (TTL of 2 was used in simulation and experiments). When the burst period propagation time ends, the node requesting for the burst transmission can start its transmission.

A burst period is divided into three slots as shown in Fig. 1. The first slot of length $\lambda$ is used by the active node to propel packets to the C-node. The second and the third slots are used by the C-node and its downstream node to forward packets to the sink. Assuming the C-node aggregates packets with aggregation ratio $\rho$, the time required to forward the aggregated packets is $\rho\lambda$. If we do not reserve slots for the C-node and its downstream node to forward their packets, when a burst period ends and all the nodes resume their transmissions, they will have little chance to forward these packets because it has to compete the channel with other nodes in the interference range again. Reserving dedicated slots for the C-node and its downstream node to transmit their packets in burst can avoid queue buildup at them.

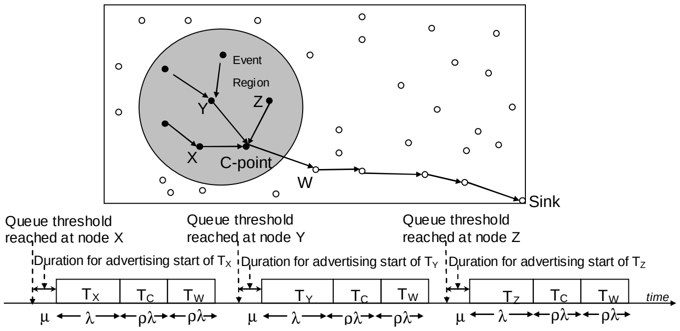


Fig. 1. **The Operation of ClearBurst:** $T_i$ indicates the duration for which a link is sending a burst to its next hop. During such a burst all nodes in the range of interference from the receiver defer their transmissions.

During a burst period, both the active node and the C-node keep announcing the progress of the burst operation by including the remaining time of the burst period in data packet headers. Any neighboring node who missed the scheduling information during the schedule advertisement time window and overhears the scheduling information freezes its transmission immediately. This further minimizes the chance of interference during burst periods.

Due to unpredictable channel conditions and unsynchronized clocks and duty cycles, some nodes may miss the scheduling information of a burst period and still try to access the channel. In order to let the active node dominate its use of the channel under potential interference from its neighbors, the active node uses smaller initial backoff and congestion window size. This helps suppress unexpected transmissions originating from the neighbors of the active node and the C-point during burst periods. The smaller initial window size also helps to minimize the overhead of initial backoff and improve channel utilization during burst periods.

When the burst period ends, all the nodes go back to pure CSMA mode to contend for the channel, and other child nodes of C-node whose queue length exceeds the threshold can start requesting another burst transmission.

## III. PERFORMANCE ANALYSIS

### A. Throughput

In this section we analytically derive throughput of CSMA and ClearBurst protocols. We adapt Bianchi's work [9] for our analysis. However, in the simulation we found that "capture" phenomenon has big impact on overall throughput when the contention is high. The capture effect is not modeled in [9]. In this section we consider the capture effect and derive the corresponding throughput.

First we need the probability that a node will transmit in an idle slot, called the transmission probability $\tau$. The backoff mechanism in MAC layer determines the transmission probability. The default MAC layer in sensor motes, e.g. Mica2, uses fixed backoff and the initial backoff window size is $16$. If the channel is busy, nodes backoff with backoff window size $32$. To simplify the analysis, we assume that the backoff window size is always $W$. Using a discrete time markov chain with $W$ states as in [9] we can easily show that in steady state, the transmission probability, $\tau$, is $\frac{2}{W+1}$.

With $\tau$, we can derive the probability that a transmission is a success or a collision. First we need to know what is the probability that a "capture" will happen if two nodes transmit at the same time. Though it is possible that the capture can still happen if three or more nodes transmit at the same time, the probability is small compared to two nodes scenario, and we ignore this case in the analysis.

A "capture" happens if a packet with stronger signal can be decoded correctly despite interference from a weaker signal. In the simulation we use two-ray ground propagation model; therefore the signal strength is inversely proportional to the square of the distance. Suppose the capture threshold is $C_t$. If nodes $a$ and $b$ transmit a packet to $s$ at the same time with the same transmission power, the packet from node $a$ can be decoded if the distance between $s$ and $b$ is at least $\sqrt{C_t}$ times than the distance between $s$ and $a$, as shown in Fig. 2.

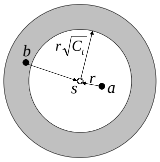


Fig. 2. If $a$ and $b$ transmit their packets to sink at the same time, $a$'s packet can still be decoded correctly at the sink due to the capture effect

We assume that $N$ nodes are uniformly distributed within a circle region with radius $R$. All nodes are within interference range of each other and the sink is at the center of the circle. Therefore if a node at distance $r$ to the sink transmits to the sink, only nodes within radius $r \times \sqrt{C_t}$ can collide with the transmission if they transmit at the same time. In a uniformly distributed deployment, there are $N \times \frac{(r\times\sqrt{C_t})^2}{R^2} - 1$ nodes in this region other than the sender. Therefore the probability that a transmission will be successfully received is

$$p = \begin{cases} (1-\tau)^{N\times\frac{(r\times\sqrt{C_t})^2}{R^2}-1} & \text{if } r < \frac{R}{\sqrt{C_t}} \\ (1-\tau)^{N-1} & \text{if } r >= \frac{R}{\sqrt{C_t}} \end{cases} \tag{1}$$

Equation 1 is the conditional probability that given a node at distance $r$ to the sink, what's the probability that this transmission is a success. To derive the success probability for entire network, we can integrate $p$ in Equation 1 from $r = 0$ to $R$, and we get $p_s =$.

$$\frac{\int_0^{\frac{R}{\sqrt{C_t}}} 2r \times (1-\tau)^{N\times\frac{(r\times\sqrt{C_t})^2}{R^2}-1} dr + (R^2 - \frac{R^2}{C_t}) \times (1-\tau)^{N-1}}{R^2} \tag{2}$$

Note that $p_s$ is the conditional probability given that at least one node is transmitting.

To compute throughput for one node, we can first compute the expected time it spent on a successful transmission, collision transmissions, and the time it sensed the channel as busy and idle. These values can be approximated as the probability of occurrence of each condition, multiplied by the time spent on that condition. We then use the expected transmitted bytes on successful transmission, which can be approximated as the probability of successful transmission multiplied by the data packet size, divided by the times spent on each condition to compute the throughput.

Therefore we need the probability and duration of each condition:

1) A node is transmitting and the transmission is a successful transmission. The probability of this condition is $P_s = \tau \times p_s$ and the duration is $T_s$ where $T_s$ is the time to transmit a data packet, ack packet, plus DIFS, SIFS time, and two propagation delays.
2) A node is transmitting but the transmission collides with others. The probability of this condition is $P_c = \tau \times (1 - p_s)$ and the duration is $T_d$ where $T_d$ is the time to transmit a data packet plus DIFS and one propagation delay.
3) The channel for the node is busy. There are two possibilities. The first is the channel is busy because of a successful transmission. The first probability is $P_{bs} = (1-\tau) \times (N-1) \times \tau \times (1-\tau)^{N-2}$ and the time is $T_s$. The second possibility is the channel is busy because of a collision. The probability is $P_{bc} = (1-\tau) \times (1-(1-\tau)^{N-1}) - P_{bs}$ and the time is $T_d$.
4) The channel is idle. The probability is just $P_i = (1-\tau)^N$ and the time is a time slot $\rho$.

Therefore the throughput for CSMA is

$$\frac{P_s T_d}{P_s T_s + P_c T_d + P_{bs} T_s + P_{bc} T_d + P_i \rho} \tag{3}$$

At C-node, we reserve few slots of the channel designated for sources, C-node, and upstream nodes of C-node, to forward the packets to the sink in a burst, as shown in Fig. 1. Assume that there is no other nodes transmitting during these reserved slots and C-node does not aggregate packets, i.e. $\rho = 1$, and the time for nodes to request for the burst period is $\mu$. We can compute the throughput of the C-node as there is only one node transmitting, i.e. $N = 1$, for the portion of that slot, which is $\frac{\lambda}{3\lambda+\mu}$. At the intermediate nodes, nodes have to contend the channel with their two-hop upstream nodes and two-hop downstream nodes, therefore there are five nodes contend for the channel. Therefore the throughput can be computed as $N = 5$. In this case, the throughput at the intermediate nodes are smaller than the throughput at C-node, therefore we can use the throughput at intermediate nodes as the system throughput.

In simulations, the data packet size is $40$ bytes, the ack packet size is $12$ bytes, and the bandwidth of the radio is $19.2Kbps$. For ClearBurst, $\lambda = 0.3$ and $\mu = 0.1$. By plugging these numbers into the equation, we can get the analytic throughput for one node in CSMA. For ClearBurst, since we reserve the channel to be used by one node, ideally there will be no contension. Therefore we can use the CSMA throughput with only one node transmitting, times the time share used by the node, which is $3/10$ in simulation.

We run simulations on CSMA and ClearBurst and compare the results with our analysis. The simulation methodology is described in Section IV. For $100m \times 100m$ event size, there are around $15$ to $30$ sources when the network is deployed with $500$ to $1000$ nodes, all are within interference range. The results are shown in Fig. 3. We can see that the throughput of CSMA drops as the network density increases, while ClearBurst remains similar across different network densities, and ClearBurst performs much better than CSMA. This confirms our claim and demonstrates the benefit of ClearBurst.

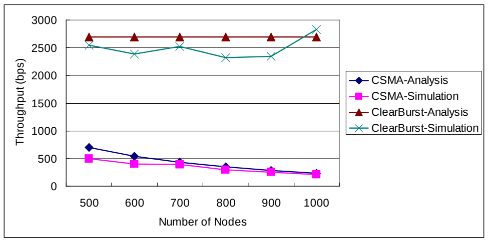


Fig. 3. Analysis and simulation results for CSMA and ClearBurst

### B. Energy Consumption

In this section, we analyze the energy consumption of CSMA and ClearBurst protocols. When the contention is high, many transmissions will result in collisions, and nodes have to retransmit these packets. Therefore, in this section we use the expected number of transmissions to successfully transmit one packet as the metric to compare ClearBurst with CSMA protocol. The number of transmissions per successful transmission is

$$E_{tx} = \frac{TX_{success} + TX_{collision} + TX_{ack}}{TX_{success}} \tag{4}$$

where $TX_{success}$ is the number of successful transmitted packets, $TX_{collision}$ is the number of collisions, and $TX_{ack}$ is the number of ACK packets. Assume that there is no collision for ACK packets, $TX_{ack} = TX_{success}$. Therefore Equation 4 becomes

$$E_{tx} = 2 + \frac{TX_{collision}}{TX_{success}} \tag{5}$$

We have computed the conditional probability $p_s$ in Equation 2. $p_s$ represents the probability of successful transmission when a node transmit. Therefore, the probability that the transmission will result in a collision is $p_c = 1 - p_s$. Accordingly Equation 5 becomes

$$E_{tx} = 2 + \frac{p_c}{p_s} \tag{6}$$

The expected number of transmissions for entire network can be approximated by $N \times E_{tx}$. For ClearBurst, when nodes are in burst transmission, only one node will transmit, and there is no collision. Therefore the expected number of transmissions for a successful transmission is two. When nodes are in CSMA duration, the expected number of transmissions is $E_{tx}$. In simulations, burst transmissions occupy $\frac{9}{10}$ of the transmission time and CSMA only accounts for $\frac{1}{10}$ of the transmission time. Therefore the expected number of transmissions for ClearBurst is $2 \times 0.9 + (2 + p_c/p_s) \times 0.1$.

| Parameter | Value |
|---|---|
| Communication range | 100m |
| Carrier sensing range | 220m |
| Channel bit-rate | 19.2Kbps |
| Initial backoff window size | 15 |
| Congestion window size | 32 |
| Queue size | 50 pkts |
| Congestion threshold | 20 pkts |
| Burst duration | 1.9902s |
| CSMA duration | 0.438575s |

TABLE I
PARAMETERS USED IN *ns-2* SIMULATION

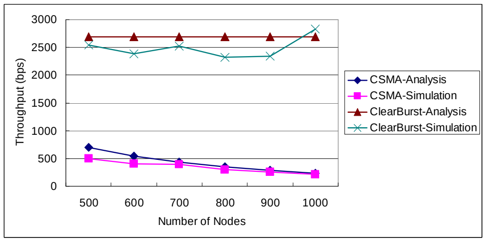


Fig. 4. Analysis and simulation results for CSMA and ClearBurst

Fig. 4 shows the analytical and simulation results. In analysis we do not consider the number of transmissions between the C-node and the sink because it depends on the distance between the sources and sink. In simulations, the number of transmissions per successful transmission is computed as the total number of transmissions for entire network, divided by the number of received packets, which includes the number of transmissions between the C-node and the sink. Therefore there are small gaps between the analysis and simulation results. However the trends are similar. When the node density increases, CSMA consumes more energy for each successfully received packet. For ClearBurst, the results are quite steady. This shows that ClearBurst not only increases the throughput, but also reduces the energy consumption.

## IV. SIMULATION

In this section, we validate the design of ClearBurst using *ns-2* (version 2.29) [12]. The parameters we used in ns-2 simulation are listed in Table II.

To demonstrate how ClearBurst works, we measure the incoming throughput at a C-point during CSMA and burst periods. As described in the protocol section, before a burst period is triggered, every sensor nodes perform pure CSMA to compete for the channel. The throughput of CSMA and burst periods are depicted in Fig. 5 as solid bars and empty bars. It can be observed that the packet incoming rates at the C-point during burst periods are 2-3 times higher than CSMA periods. This is because ClearBurst not only mitigates contentions but also eliminates the hidden terminal problem by shutting off potential interfering nodes during burst periods. In Fig. 5, before the next CSMA period starts, there is a period in which the packet arriving rate is zero. This is because burst period is divided into three sub periods, and the second and the third sub periods are used by the C-point and its nexthop to move packets out of the event region.

To study the performance of ClearBurst, extensive simulations are conducted using random topologies with various densities, event sizes and data rates. Performance metrics include throughput, energy tax, and latency. Energy tax is defined as $(TX_{data} + TX_{ack})/R$, where $R$ is the number of packets received at the sink, $TX_{data}$ is the total number of transmissions for data packets and $TX_{ack}$ is the total number of transmissions for ACK packets. Energy tax represents the average number of transmissions required to forward a packet to the sink. Throughout the simulations, we assume that all the traffic generated by the sensors in the event region are aggregated at a C-point before they are forwarded to the sink.

### A. Network Densitiy

We generate six sensor networks of area $1000m$ by $1000m$ to evaluate performance for different network densities, with $500$ to $1000$ nodes uniformly distributed in the network. Node $0$ is at the bottom left corner and serves as the sink. Sensors within the event region generate traffic at a constant rate of 5 pkts/s. For each simulation, $30$ static events with $100m$ radius are randomly generated. The average throughput, energy tax and latency with $95\%$ confidence intervals are plotted in Fig. 6(a), 6(b), and 6(c) respectively.

As network density increases, more and more sensors are located in the event region. This increases the channel contention and results in more collisions which leads to packet drops in CSMA. Even worse, the aggregation point has no chance to forward packets accumulated in its queue. Therefore even when packets are successfully delivered to the C-point, only a few of them can be forwarded toward the sink. Therefore, CSMA has low average throughput of $1.1 pkts/s$. CSMA's throughput decreases by $65\%$ (from 1.7 pkts/s to 0.6 pkts/s) as network density increases. In addition, in Fig. 7(b), high energy tax can also be observed in high network density scenarios in CSMA. This indicates more packet transmissions are wasted due to the increased contention. By observing the queue occupancy at the aggregation node, we found the waste is due to contention induced queue overflow which causes most of the packets dropped at the C-point. The intensive contention also results in high latency, which is shown in Fig. 6(c), because packets must wait a long time in the queue of the C-point.

In contrast, the throughput of ClearBurst keeps steady in the range of $5.5$ pkts/s to $6$ pkts/s, which is 6 times higher than CSMA in average. Furthermore, energy tax also keeps steady in the range of $35$ to $40$. This proves that ClearBurst can achieve higher throughput and is more energy efficient than CSMA.

### B. Event Radius

To evaluate the impact of event size, we fix the network density at $1000$ nodes and vary event radius from $50m$ to $120m$. Sensors in the event region report their readings to the aggregation node at the rate of 5 pkts/s. Fig. 7(a), 7(b), and 7(c) show the throughput, energy tax and latency of CSMA and ClearBurst. In this set of simulation, we can see a clearer trend of performance degradation of CSMA. When the event radius is as small as $50m$, in average there are only six sensor nodes in the event region. With a congestion window of size $32$, CSMA can arbitrate the channel access efficiently. Therefore its throughput is only 1 pkts/s lower than ClearBurst. However CSMA has lower latency when the event radius is smaller than $60m$. This is due to the fact that in the design of ClearBurst, sensors must withhold their transmission during burst durations which incurs delay.

However, as the event radius grows to $120m$, CSMA's throughput drops dramatically from $5.3$ pkts/s to $0.6$ pkt/s and energy tax grows from $90$ to $250$. By contrast, ClearBurst's throughput stays above $5$ pkt/s and energy tax keeps steady in the range of $35$-$40$. When the event radius is greater than $70m$, CSMA starts to suffer from the intensive contention and has higher latency than ClearBurst.

### C. Source Rates

In this set of simulation, network density is fixed at $1000$ nodes and event radius is fixed at $100m$. Source rate varies from $0.5$ pkt/s to $5$ pkt/s. In Fig. 8, we can observe similar results as in Fig. 6 and 7. However, when the source rate is as low as $0.5$-$1$ pkt/s, CSMA is as efficient as ClearBurst in terms of throughput and energy tax and has lower latency than ClearBurst. This is due to that there are only $31$ sensors in the event region and they generate traffic at a low rate (less than $1$ pkt/s). CSMA can efficiently arbitrate the transmission among sensor nodes. However, as the source rate increases, CSMA's performance decreases.

In Fig. 8, we can also observe the throughput decrease in ClearBurst when the source rate increases from $0.5$ pkts/s to $1$ pkt/s. The reason is when a burst duration starts, some sensors can remain active because it does not hear the scheduling information and thus do not shut off their transmissions. This causes interference in the beginning of burst durations and decreases throughput.

### D. Moving Events

Where event is moving, ClearBurst can still be applied in protocols such as DCTC [10]. In DCTC, a data collection tree is dynamically constructed and reconfigured to tract the event and all sensors send event status to the root of the tree where the final report is generated and forwarded to the sink. Once the C-point is elected, it automatically serves as a coordinator and arbitrate the use of burst durations among its child sensor nodes.

We simulate moving events with $100m$ event radius and four-way random motion. All the sensors that can detect the event generate event status update at the rate of $5$ pkts/s.

Fig. 9 shows the throughput, energy tax, and latency under various event moving speeds. When the event speed is low, similar performance can be observed as in static events. However, as event speed increases, CSMA's throughput starts to increase and converges to that of ClearBurst. This is because under high event speed, C-point changes frequently and the congestion around the C-point only lasts for a short period of

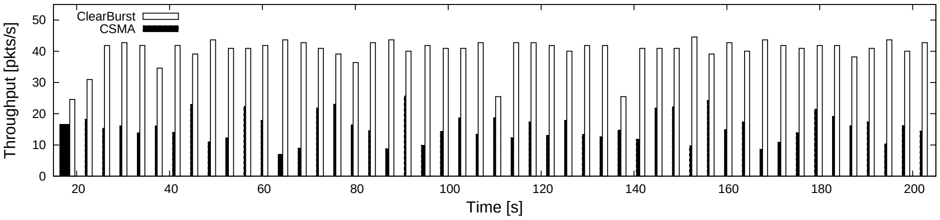
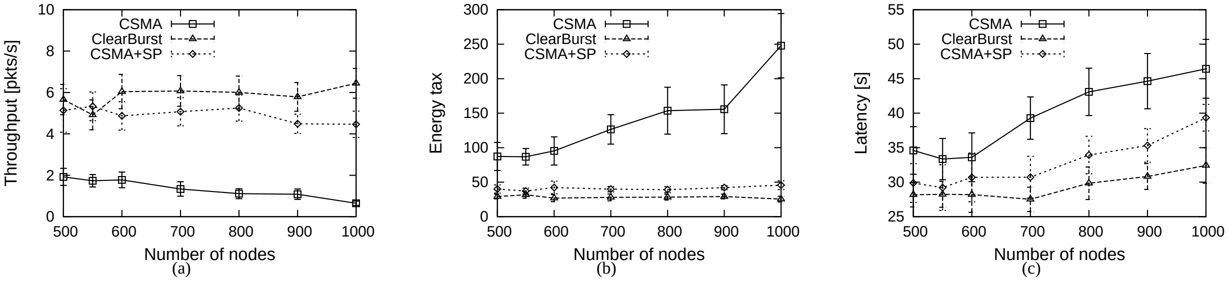
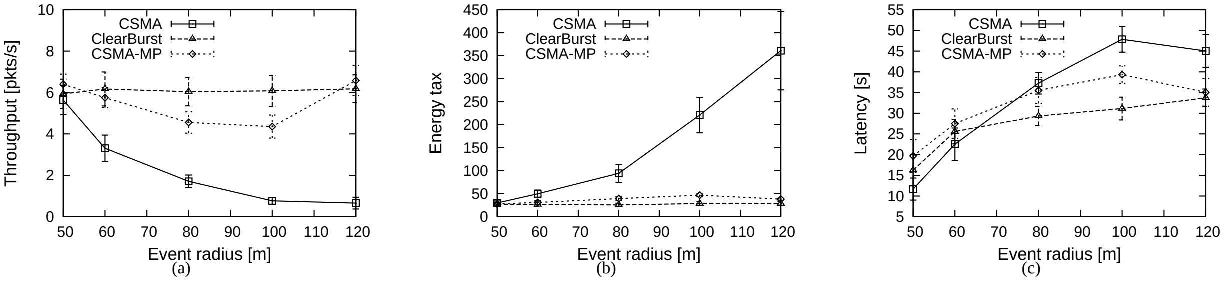


Fig. 5. Incoming throughput measurement for CSMA and ClearBurst at a C-point.



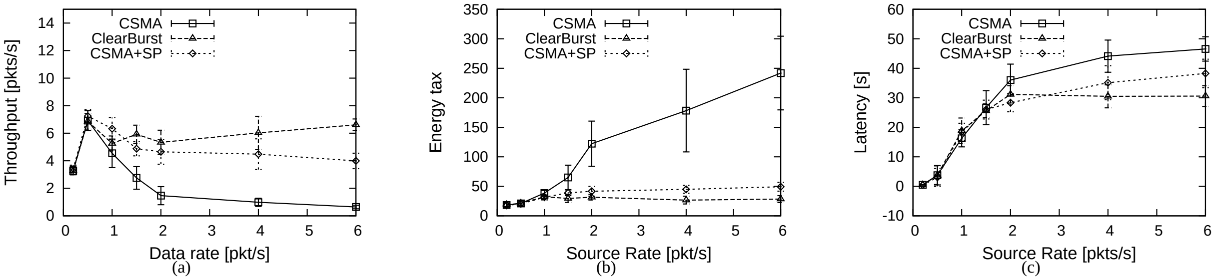
(a) (b) (c)

Fig. 6. Performance evaluation of various network densities



(a) (b) (c)

Fig. 7. Performance evaluation of various event sizes



(a) (b) (c)

Fig. 8. Performance evaluation of various source rates

time. Furthermore, when a previous C-point is still aggregating and sending report to the sink, a new C-point could be elected and start sending report to the sink. These factors lead to the increased throughput for CSMA. However, traffic originated from different C-points will interfere with each other as they flow toward the sink, and this offsets the benefits of applying ClearBurst. Fig. 10(b) and 10(c) show similar results for energy tax and latency. When the event speed is low, CSMA's energy tax is $5$ times higher than that of ClearBurst and latency is also $25\%$ higher. As the speed increases, CSMA's energy tax converges to that of ClearBurst. However, ClearBurst incurs higher latency than CSMA when event speed is high.

### *E. Queue Threshold*

ClearBurst uses a queue occupancy threshold to determine when to trigger and perform ClearBurst. To understand how different threshold values impact its performance, simulations are conducted to evaluate the performance of ClearBurst under

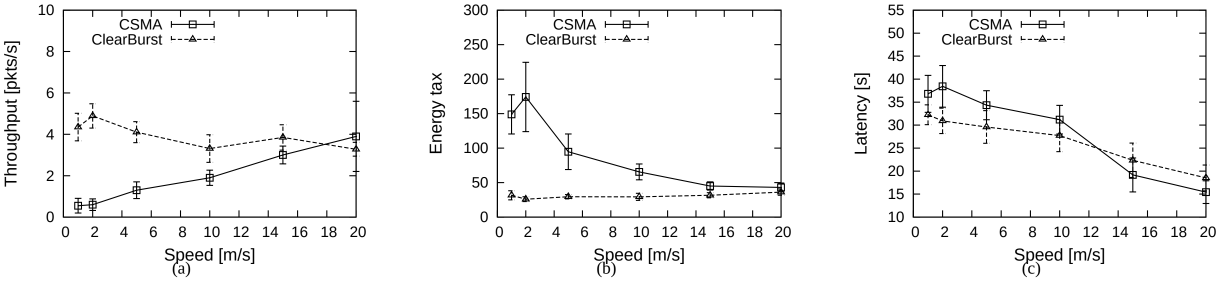


Fig. 9. Performance evaluation of moving events

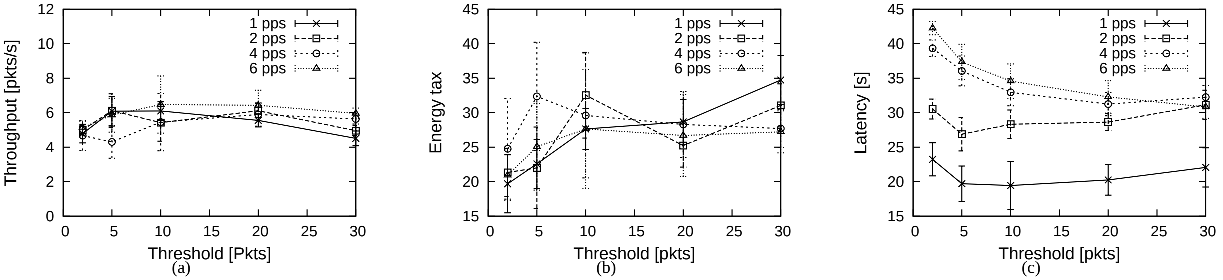


Fig. 10. Performance evaluation of various thresholds

various threshold values. We varied the threshold from $2$ packets to $30$ packets. As shown in Fig. 10(a) and 10(b), threshold values do not have significant impact on throughput and energy tax. However, Fig. 10(c) shows a clear trend that if threshold hold is greater than $40$ packets, latency increases by $25\%$.

### *F. Fairness*

## V. EXPERIMENTS

We implement ClearBurst on TinyOS [13] to evaluate its performance in real environment. Fig. 12 shows the architecture of ClearBurst TinyOS implementation [13]. To support the functions required by ClearBurst, we extended the MacControl interface to include the following commands and event.

1) **SetPriority**: Specify the priority. When MAC is in high priority, it uses smaller initial backoff and contention window size.
2) **SetSlotLength**: Set the length of the burst period needed to flush the number of BURST_THRESHOLD packets in the queue.
3) **CountCompleted**: An event used by the MAC layer to inform ClearBurst module that the specified burst period or CSMA period has ended.

A circular queue of size $32$ is implemented for ClearBurst. The ClearBurst module interacts with the circular queue through the CirQueueControl interface. CirQueueControl interface provides three commands which include Enqueue, Dequeue, and Length. This circular queue serves as the TX queue which is shared by all the applications. Messages generated locally and packets being routed through a node are sent to the queue first. The ClearBurst module then pops one packet from the queue each time and transmits it to the next hop.

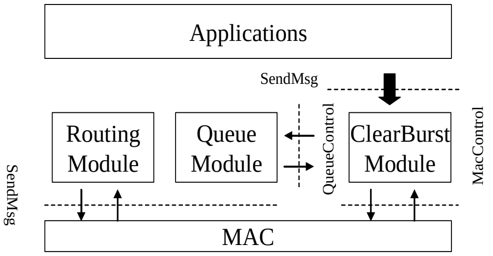
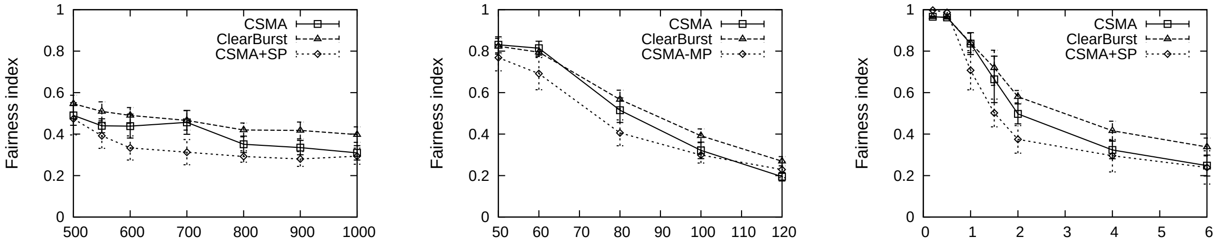


Fig. 12. TinyOS implementation.

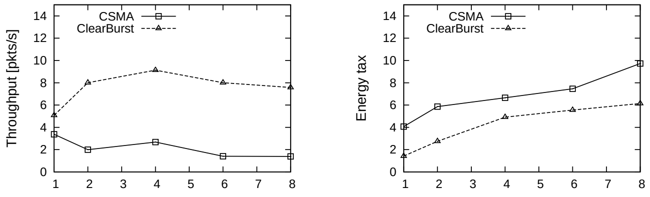
The ClearBurst module continuously monitors the queue length. If it exceeds the congestion threshold, ClearBurst intervenes packets transmission. Before sending a packet to the MAC layer, the ClearBurst module calls *MacControl.SetSlotLength()* command to pass the remaining time of the burst period to the MAC layer. The MAC layer stamps the remaining time in the header right before the packet is to be transmitted. Every time *SpiByteFifo.dataReady()* interrupt handler in the MAC are executed, the remaining time of current burst period is decremented. When the time reaches zero, the MAC layer informs ClearBurst module by signaling a *MacControl.CountComplete* event.

Experiments are conducted on Kansei testbed [14], [15] with $49$ nodes in grid topology. The sink is located at the bottom right corner. Eight nodes at the top left corner periodically send a packet to the C-node. In the experiments, C-nodes are manually selected.



(a) (b) (c)

Fig. 11. Fairness evaluation of various network density, event radius and reporting rate



(a) (b)

Fig. 13. Performance evaluation of various source rates on Kansei testbed.

As shown in Figs. 13(a) and 13(b), ClearBurst achieves four times higher throughput than TinyOS's CSMA MAC and yet is more energy-efficient. The results are similar to the results presented in the previous section. However, it clearly demonstrates that, even with a small network, the many-to-one traffic pattern in sensor networks has a severe impact on data delivery. Thus, transmission in the neighborhood of C-nodes must be coordinated, which in turn validates the design of our protocol.

## VI. CONCLUSIONS

This paper addresses the overload problem and provides a solution to improve system throughput and reduce energy consumption. The proposed MAC layer solution makes use of bursty transmissions with low-overhead local advertisements to avoid contention during the bust-periods. Using extensive simulations we observe that the performance of our approach is better than CSMA, with an increasing performance gap as the network gets overloaded (higher nodes density and/or larger event size). These observations are also supported by the experiments on the *Kansei* testbed on different data rates. We conclude that our proposed approach is highly suited for sensor networks for data collection applications.

## REFERENCES

[1] G.-S. Ahn, S. G. Hong, E. Miluzzo, A. T. Campbell, and F. Cuomo, "Funneling-MAC: A Localized, Sink-oriented MAC for Boosting Fidelity in Sensor Networks," in *Proc. of SENSYS*, 2006, pp. 293–306.

[2] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," in *Proc. of SENSYS*, 2005, pp. 90–101.

[3] C. Lim, H. Luo, and C.-H. Choi, "RAIN: A Reliable Wireless Network Architecture," in *Proc. of ICNP*, 2006, pp. 228–237.

[4] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks," in *Proc. of SENSYS*, 2003, pp. 266–279.

[5] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating Congestion in Wireless Sensor Networks," in *Proc. of SENSYS*, 2004, pp. 134–147.

[6] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis, "Interference-aware Fair Rate Control in Wireless Sensor Networks," in *Proc. of SIGCOMM*, 2006, pp. 63–74.

[7] I. Rhee, A. Warrier, J. Min, and L. Xu, "Drand:: distributed randomized tdma scheduling for wireless ad-hoc networks," in *Proc. of MOBIHOC*, 2006, pp. 190–201.

[8] G. Bianchi, L. Fratta, and M. Oliveri, "Performance Evaluation and Enhancement of the CSMA/CA MAC Protocol for 802.11 Wireless LANs," in *Proceedings of 7th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 2, Octover 1996, pp. 392–396.

[9] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Func," in *IEEE Journal on Selected Areas in Communications*, vol. 18, March 2000, pp. 535–547.

[10] W. Zhang and G. Cao, "DCTC: Dynamic Convoy Tree-Based Collaboration for Target Tracking in Sensor Networks," pp. 1689–1701, 2004.

[11] K. Xu, M. Gerla, L. Qi, and Y. Shu, "Enhancing TCP Fairness in Ad Hoc Wireless Networks Using Neighborhood RED," in *Proc. of MOBICOM*, 2003, pp. 16–28.

[12] "Ns-2 homepage: http://www.isi.edu/nsnam/ns/."

[13] "Tinyos homepage: http://www.tinyos.net/."

[14] E. Ertin, A. Arora, R. Ramnath, M. Nesterenko, V. Naik, S. Bapat, V. Kulathumani, M. Sridharan, H. Zhang, and H. Cao, "Kansei: A Testbed for Sensing at Scale," in *Proceedings of the 4th Symposium on Information Processing in Sensor Networks (IPSN/SPOTS track)*, 2006.

[15] A. Arora, E. Ertin, R. Ramnath, W. Leal, and M. Nesterenko, "Kansei: A High-Fidelity Sensing Testbed," in *IEEE Internet Computing, special issue on Large-Scale Sensor Networks*, March 2006.