This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2008 proceedings.

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 *ns-2*.
- 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

This material is based upon work supported by the National Science Foundation under Grants CNS-0546630 (CAREER Award), CNS-0721434, CNS-0721817 and CNS-0403342. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

1525-3511/08/$25.00 ©2008 IEEE

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 λ 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 ρ, the time required to forward the aggregated packets is $\mu = \rho\lambda$. 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.

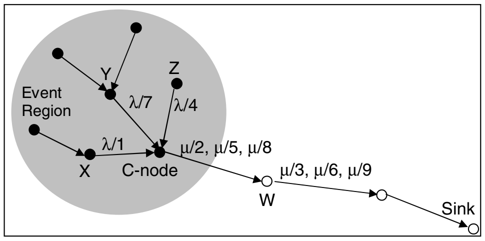


Fig. 1. **The Operation of ClearBurst:** t/i on each link indicates the duration t at time slot i for which link to send a burst to its next hop. During such a burst all nodes in the range of interference from the receiver defer their transmissions.

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-node 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 transmits in an idle slot, called the transmission probability τ. 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, τ, is $\frac{2}{W+1}$.

With τ, we can derive the probability that a transmission is a success or a collision. Second, 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 capture can still happen if three or more nodes transmit at the same time, the probability is relatively small. And we ignore this case in the analysis.

A "capture" happens if a packet with stronger signal can be correctly decoded at the receiver in the existence of 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 longer than the distance between s and a, as shown in Fig. 2.

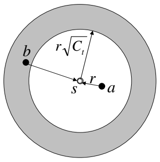


Fig. 2. If a and b transmit their packets to s at the same time, a's packet can still be correctly decoded at s due to the capture effect

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 receiver s is at the center of the circle. If we observe a tagged node whose distance to s is r. If $r < \frac{R}{\sqrt{C_t}}$, only nodes within radius $r \times \sqrt{C_t}$ can collide with its transmission. In a uniformly distributed deployment, there are $N \times \frac{(r\times\sqrt{C_t})^2}{R^2} - 1$ such nodes other than the tagged node. If $r >= \frac{R}{\sqrt{C_t}}$, all the other nodes can collide with the tagged node. Therefore, the probability for a node to successfully transmit a packet is

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

Equation 1 is the conditional probability of a successful transmission given that a sender is at distance r to its receiver. To derive the marginal success probability, we can integrate p defined in Equation 1 from $r = 0$ to R, and we get $p_s =$

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

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

To compute throughput for an individual node, we first compute the expected time it spends on a successful transmission, a collision, and the time the channel is sensed 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 of a successful transmission, which can be approximated as the probability of successful transmission multiplied by the data packet size, divided by the sum of the times spent on each condition to compute the throughput.

Therefore we need the probability and duration of each of the condition:

1) A node transmits 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 transmits 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) A node backsoff due to busy channel. There are two possibilities. First, the channel is busy because of a successful transmission. The probability is $P_{bs} = (1 - \tau) \times (N-1) \times \tau \times (1-\tau)^{N-2}$ and the time is T_s. Second, 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. In such case, the probability is simply $P_i = (1-\tau)^N$ and the time is a time slot ρ.

Therefore the throughput of a node operating a CSMA MAC is

$$\frac{P_s D}{P_s T_s + P_c T_d + P_{bs} T_s + P_{bc} T_d + P_i \rho} \quad (3)$$

where D is the packet size.

ClearBurst reserves a few slots of the channel for sources, the C-node, and downstream nodes of the 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 the C-node does not aggregate packets, i.e. $\rho = 1$, and the length of schedule advertisement time window is ω. We can compute the throughput of the C-node as there is only one node transmitting, i.e. $N = 1$, multiplied by the fraction of its slot to the entire epoch of burst transmission, which is $\frac{\lambda}{3\lambda+\omega}$. However, for those intermediate nodes on the path from the C-node to the sink, they have to contend the channel with their two-hop upstream nodes and two-hop downstream nodes. Therefore, the throughput of intermediate nodes can be computed as $N = 5$, which is smaller than the throughput of the C-node. Thus, the system throughput of ClearBurst is equal to the throughput at intermediate nodes.

In simulations, the data packet size is 40 bytes, the ack packet size is 12 bytes, and the bandwidth of the radio is $19.2Kbps$. By plugging these numbers into Equation 3, we can get the analytic throughput of CSMA. For ClearBurst, since the system throughput equals the throughput of intermediate nodes, we use $N = 5$ and Equation 3 to compute its analytic throughput.

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 performs much better than CSMA. This confirms our claim and demonstrates the benefit of ClearBurst.

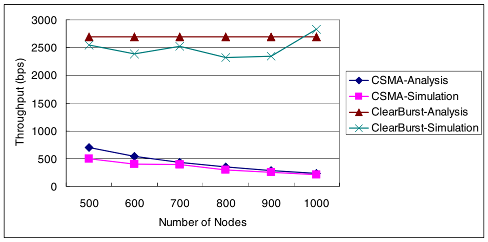


Fig. 3. Analysis and simulation results for CSMA and ClearBurst

B. Energy Consumption

In this section, we analyze the energy consumption of CSMA and ClearBurst protocols. When contention is intensive, the probability of collision is high. And a node has to retransmit the packet when collision happens. Therefore, we use the expected number of transmissions to successfully transmit one packet as the metric to compare the energy efficiency of ClearBurst and CSMA. The metric called normalized number of transmissions is defined as:

$$E_{tx} = \frac{TX_{success} + TX_{collision} + TX_{ack}}{TX_{success}} \qquad (4)$$

where $TX_{success}$ is the number of successfully 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

$$E_{tx} = 2 + \frac{TX_{collision}}{TX_{success}} \qquad (5)$$

In Equation 2, we have computed p_s, which represents the conditional probability of a successful transmission given that a node transmits. The conditional probability of a collision is therefore $p_c = 1 - p_s$. Accordingly, Equation 5 becomes

$$E_{tx} = 2 + \frac{p_c}{p_s} \qquad (6)$$

The expected number of transmissions for entire network can be approximated by $N \times E_{tx}$. For ClearBurst, when nodes are in burst transmission mode, only the C-node can transmit, and there is no collision. Therefore the expected number of transmissions for a successful transmission is two. When nodes are in CSMA mode, the expected number of transmissions is simply E_{tx}. In simulations, since burst transmissions occupy $\frac{9}{10}$ of the simulation time and CSMA accounts for $\frac{1}{10}$ of the simulation time, 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

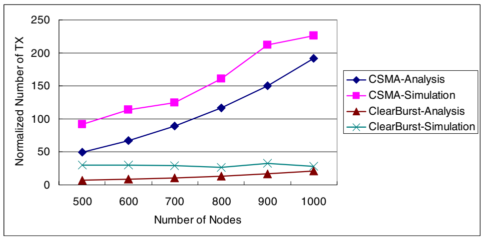


Fig. 4. Analysis and simulation results for CSMA and ClearBurst

Fig. 4 shows the analytical and simulation results. In analysis, we do not consider the transmissions of intermediate nodes that are on the path from the C-node to the sink because it depends on the distance between the C-node and the sink. However, in simulations, all transmissions in the network are considered for computing E_{tx}. Therefore there are small gaps between the analysis and simulation results. However the trends are similar. When node density increases, CSMA consumes more energy for each successfully received packet.

IV. SIMULATION

To study the performance of ClearBurst, we use *ns-2* to conduct extensive simulations using random topologies with various node densities, event sizes and source 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. Table IV shows the parameters we used in the simulations. We compare the performance of ClearBurst with CSMA and CSMA+SP. In CSMA, all data packets are forwarded to the same C-node as in ClearBurst before they are forwarded to the sink. Whereas, in CSMA+SP, data packets follow shortest-path routes. We present results for varying network density and event radius. Detailed simulation results for varying source rates, event speed, and queue thresholds can be found in the technical report [12].

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. Sink is at the bottom left corner. 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. 5(a), 5(b), and 5(c) respectively.

As network density increases, more sensors are in the event region. This increases the channel contention and results in more collisions which leads to packet drops in CSMA. Even worse, the C-node has little chance to forward packets accumulated in its queue. Therefore, even though packets are successfully delivered to the C-node, only a few of them can be forwarded toward the sink which results in low throughput and high letency in CSMA as shown in Fig. 5(a) and 5(c). In addition, it can be observed in Fig. 5(b) that CSMA has high energy tax when node density is high. This indicates more packet transmissions are wasted. By observing the queue occupancy at the C-node, we found the waste is due to contention induced queue overflow. Similar results can also be observed in CSMA+SP. Although the throughput of CSMA+SP is higher than CSMA, it is still 20% lower than ClearBurst. Furthermore, allowing packets to go through different paths reduces the chance of early aggregation. And packets along different paths have to compete for the channel, which results in 50% higher energy consumption and latency than ClearBurst.

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 $120m$. Results are presented in Fig. 6(a), 6(b), and 6(c). In this set of simulation, we can see a clear trend of performance degradation of CSMA. When the event radius is small, CSMA can arbitrate the channel access efficiently. Thus, it has similar performance as ClearBurst. As the event radius grows, CSMA's throughput drops and energy tax and latency increase dramatically. In contrast, ClearBurst's throughput and energy tax remain steady. We also observe that when event radius reaches $120m$, the performance of CSMA+SP becomes comparable to ClearBurst. This is because when event radius is large, a single C-node can cause long detour paths which can lead to lower throughout, higher latency and energy consumption. This problem can be optimized by electing multiple C-nodes. However, the burst transmission of nearby C-nodes must be scheduled carefully so they do not interfere with each other. And we leave it as future work.

V. EXPERIMENTS

We implement ClearBurst on TinyOS [13] to evaluate its performance in real environment. Fig. 7 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.

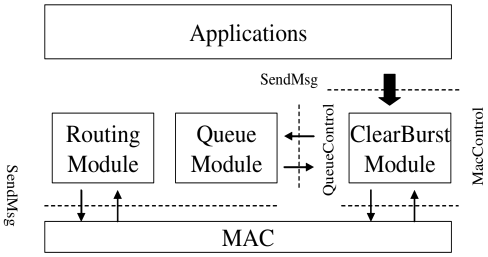


Fig. 7. TinyOS implementation.

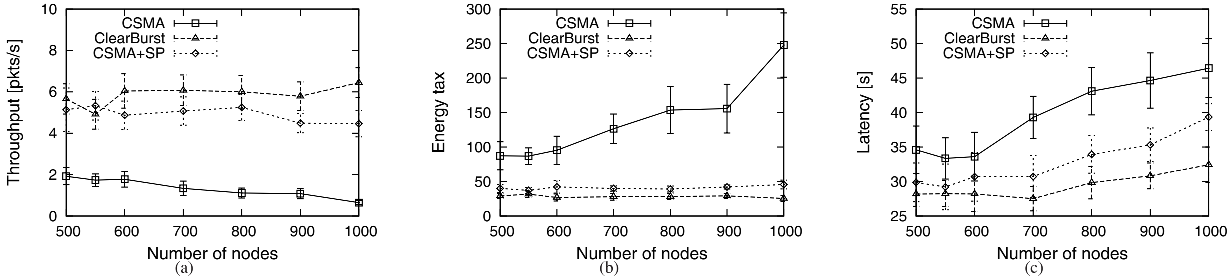
1) **SetPriority**: Specify the priority. When MAC is in high priority, it uses smaller initial backoff and contention window size.
2) **SetSlotLength**: Set the length of the burst period needed to flush *BURST_THRESHOLD* packets in the queue.
3) **SlotExpired**: An event used by the MAC layer to inform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

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.SlotExpired* event.

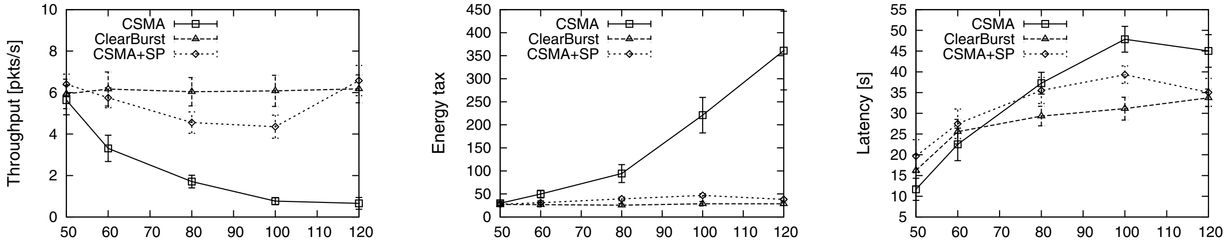
Experiments are conducted on Kansei testbed [14], [15] with 49 nodes in grid topology. The sink is located at the bottom right corner. Eight nodes at the top left corner periodically send a packet to the C-node. In the experiments, C-nodes are manually selected.

As shown in Figs. 8(a) and 8(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-node must be coordinated, which in turn validates the design of our



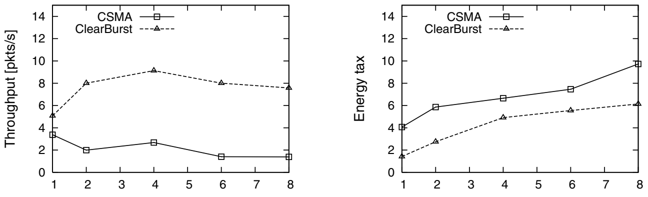
(a) (b) (c)

Fig. 5. Performance evaluation of various network densities



(a) (b) (c)

Fig. 6. Performance evaluation of various event sizes



(a) (b)

Fig. 8. Performance evaluation of various source rates on Kansei testbed.

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 burst 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] R.-S. Liu, K.-W. Fan, and P. Sinha, “ClearBurst: Burst Scheduling for Contention-Free Transmissions in Sensor Networks,” in *Technical report OSU-CISRC-9/07-TR68*, 2007.

[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.