Localized QoS-Aware Media Access Control in High-Fidelity Data Center Sensing Networks

Renjie Huang Wen-Zhan Song Mingsen Xu Behrooz Shirazi

Sensorweb Research Laboratory
Washington State University, Vancouver, WA 98686
Email: {renjie_huang, songwz, mingsen_xu, shirazi}@wsu.edu

***Abstract*—Low-cost non-intrusive wireless sensor network is ideal for high-fidelity data center monitoring and helping improve energy efficiency. However, the stringent quality of service (QoS) requirements such as real-time data delivery together with the dynamic nature of low-power wireless communication render traditional MAC protocols inapplicable to high-fidelity data center sensing networks. In this paper, we present RoMac, a localized QoS-aware TDMA MAC protocol for high-fidelity data center sensing networks. The key contributions of RoMac include: (1) it provides collision-free communication and persistent QoS guarantee such as constant packet jitter and bounded latency; (2) it is resilient to topology dynamics and truly traffic adaptive in dynamic environments. To our best knowledge, this is the first TDMA MAC protocol that can achieve the resilience to dynamic topologies; (3) its localized design eases the need of scheduling message exchange beyond the initialization phase. We present an implementation of the protocol using the TinyOS operating system running on the iMote2 sensor mote platform. Our experiment result shows that RoMac can achieve the adaptability to dynamic changes in data collection sensor networks and provide persistent QoS support in terms of timeliness, packet jitter and fairness.** [1]

***Keywords*-QoS-Aware, Media Access Control, Data Sensing Netowrks**

I. INTRODUCTION

Energy consumption is a critical concern for IT industry as the cost of operating data centers increases due to the growing use of computing devices and rising energy costs. Low-cost non-intrusive wireless sensor network is ideal for monitoring temperature, humidity, power consumption, air flow and wetness conditions in data centers and helps improve energy efficiency. However, wireless sensor networking faces significant challenges in a data center environment. The application requirements of high data fidelity and quality of service (QoS) such as real-time delivery together with the dynamic nature of low-power wireless communication render traditional MAC protocols inapplicable for data center sensing networks. Firstly, such networks need to provide sensing with high temporal as well as spatial fidelity to provide visibility into thermal dynamics. The sampling rate is sub-minute or even higher when troubleshooting hot spots. In addition, the network is typically of large scale with high node density. Even a single data center room may require hundreds of wireless sensors [1]. The number of nodes in the communication neighborhood can be very large, thus uncoordinated wireless transmissions can lead to severe channel contention and congestion collapse drastically reducing the network's usable capacity. A recent data center monitoring deployment shows that 50% to 65% of the sensors can interfere with each other [1]. Secondly, the sensory data of data centers demands real-time delivery not only for diagnosing problems but also for improving the data center's energy efficiency. However, CSMA-based MAC protocols utilize carrier sense and backoff to deal with channel contention and thus the data delivery often experiences long latency in such dense networks and do not meet the real-time constraints. Thus CSMA-based MAC protocols are inefficient in such environments.

While traditional multi-hop TDMA MAC protocols [2]–[4] in wireless data collection sensor networks can provide collision-free wireless communications, they are not designed for provisioning QoS support. In addition, they are vulnerable to network dynamics and unable to provide persistent QoS guarantee. Sensor networks in data center environments often experience frequent topology changes because of time-varying channel conditions [5], physical environmental changes, and node relocation. For example, co-located 802.11 networks and the large number of metallic obstacles can result in a harsh RF environment [1]. The dynamics can lead to challenging problems. Firstly, in data collection networks an intermediate node has to forward more data if new nodes join its subtree. That will cause congestion due to the mismatch between the new bandwidth demand and the old bandwidth assignment. Secondly, upon the occurrence of topology dynamics, a schedule that ensures conflict-free transmission may become invalid and lead to conflicts. It is expensive for conventional multi-hop TDMA MAC protocols to handle dynamic topology changes, which possibly requires a global change to avoid scheduling conflicts in 2-hop neighborhood. Periodic beaconing for slot assignment is a common strategy for addressing topology dynamics. However, it is costy as well as unresponsive due

---

[1] Our research work is supported by NSF-CNS-0914371, NSF-CNS-0953067 and Hong Kong ITF.

978-1-4244-7614-5/10/$26.00 ©2010 IEEE

to the latency for the slot assignment update to be reliably disseminated. Thus those TDMA MAC protocols cannot provide persistent QoS guarantee.

In this paper, we propose RoMac, a localized QoS-aware TDMA MAC protocol for high-fidelity data center sensing networks. Departing from traditional MAC protocols, the primary goal of RoMac is to achieve persistent QoS in terms of data reliability and timeliness in dynamic and challenging environments. QoS-aware MAC protocol is not only important for data center sensing networks, but also critical for a variety of cyber physical systems where real-time interaction with the physical world is crucial. For example, smart energy-harvesting wind turbine requires real-time sensor data to adjust turbine blades to rapidly respond to changing wind conditions for efficiency. Our contributions include:

- With RoMac, channel access is well regulated to ensure data reliability and persistent QoS. It provides bounded delay and constant jitter for each sensor's data flow.
- RoMac achieves resilience to topology changes and is truly *traffic adaptive* without facing problems of scheduling conflict and congestion. To our best knowledge, this is the first TDMA MAC protocol that can achieve the resilience to dynamic topologies.
- The localized design eases the need of scheduling message exchange between sensor nodes beyond the initialization phase (this is even not necessary in normal cases), thus the control overhead is almost zero.

The rest of the paper is organized as follows. In Section II, we review existing MAC protocols for wireless sensor networks. In Section III we introduce the design of RoMac by considering various dynamics in real network environments. Section IV presents the implementation of RoMac in the TinyOS operating system. We then evaluate the protocol performance in Section V. Finally, Section VI summarizes the paper.

II. RELATED WORKS

Low Duty Cycle MAC Protocols. Existing sensor network MAC protocols mainly address energy conservation in low-duty-cycle applications. Synchronous approaches such as S-MAC [6], T-MAC [7], and SCP-MAC [8] synchronize neighboring nodes in order to align their active or sleeping periods. Neighboring nodes exchange packets only within the common active time, allowing sensor nodes to sleep for most of the time within an operational cycle. Asynchronous duty cycle MAC protocols such as B-MAC [9] and X-MAC [10] employ an adaptive preamble sampling scheme to reduce duty cycle and minimize idle listening. RI-MAC [11] is a receiver-initiated asynchronous duty cycle MAC protocol that improves throughput and packet delivery ratio while maintaining power efficiency. While the primary goal is energy conservation, those low duty cycle MAC protocols typically provide poor QoS performance in terms of delivery latency due to the time cost to achieve rendezvous between the sender and the receiver. By contrast, for RoMac the primary concern is data reliability and delivery timeliness instead of energy efficiency since it can take advantage of the energy supply from data servers.

Multihop TDMA MAC Protocols. In recent years multihop TDMA MAC protocols for data collection sensor networks have been well studied in the literature. Z-MAC [2] uses CSMA as the baseline MAC scheme, but uses a TDMA schedule as a hint to enhance contention resolution. Z-MAC does not consider the many-to-one traffic pattern in data collection network, hence it is not really traffic-driven. Observing the funneling effect that the nodes closer to the sink experience heavy congestion in high-rate data collection networks, Funneling-MAC [3] uses TDMA scheduling in the intensity region to mitigate the funneling effect, while keeping CSMA in the rest of the network to provide flexibility. Its limitation results from the assumption that the sink node has a relatively longer transmission range and can manage the TDMA scheduling in the intensity region. TreeMAC [4] proposes a localized TDMA MAC protocol for addressing the funneling effect. It makes use of the unique tree-like topology characteristics of data collection networks and let parent nodes proportionally assign bandwidth for their child nodes based on their relative bandwidth demand. However, both Funneling-MAC and TreeMAC are sensitive to route changes. The occurrence topology dynamics can cause mismatch between the bandwidth demand and the assignment of some sensor nodes resulting in congestion. The reason is that a node has to forward more data if new nodes join its subtree. While in TreeMAC parent nodes periodically update bandwidth assignment for their children, there exists latency for the slot assignment update to be reliably disseminated. In addition, most of those TDMA protocols are based on graph coloring; that is, in a multihop network nodes are scheduled so that no two nodes are assigned the same time slot within a two-hop neighborhood. However, a transmission schedule that ensures conflict-free transmission becomes invalid when the network topology changes resulting in transmission conflicts. By contrast, RoMac is truly localized and traffic adaptive without facing those problems. It is a localized QoS-aware TDMA MAC protocol which is resilient to topology changes. More importantly, it can support QoS guarantee on packet jitter and delivery timeliness and is well suited for high-fidelity data center sensing networks which require real-time delivery for diagnosing problems and improving energy efficiency.

III. ROMAC PRINCIPLES AND DESIGN

A. Basic Ideas

TDMA has been used in the wireless cellular systems such as Global System for Mobile Communications (GSM) for voice service. It allows users to share the same frequency channel by dividing the signal into different time slots.

Each cell phone gets a fixed time slot number per frame, and transmits data in rapid succession. Thus, the packet delay and jitter is constant to guarantee QoS. However, it is designed for a one-hop wireless network, and does not deal with many challenges that TDMA faces in multi-hop networks. Firstly, in multi-hop environments, intermediate nodes need to forward data for nodes further away from the sink; therefore, the network bandwidth assignment scheme should consider the bandwidth demand for the forwarding data. Secondly, different from one-hop TDMA, multi-hop TDMA protocols have to address 2-hop interference known as the hidden and exposed terminal problem. Thirdly, network dynamics also poses great challenges on multi-hop TDMA, because the originally interference-free schedule may be invalidated by the topology changes.

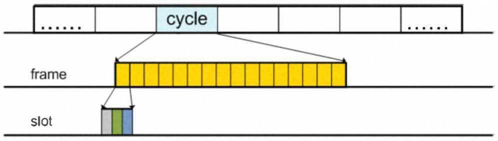

Figure 1. Relationship among cycle, frame and slot: 1 cycle = N frames and 1 frame = 3 slots.

We first present an intuitive example to illustrate how RoMac addresses those problems in data collection sensor networks. In a multi-hop network each source node generates a data flow of the same bit-rate, that is they have the same bandwidth demand for their own data (here we assume equal link reliability). Thus we divide each time cycle into frames, and each source node is assigned one frame for its own data. Assuming the network nodes IDs are $\{1, 2, \cdot\cdot, N\}$, we assign node i with the ith frame. For example, in Figure 2 node 4 is assigned the 4th frame in each cycle. To address two-hop interference, we further divide each frame into 3 slots $\{0,\ 1,\ 2\}$. Figure 1 illustrates the relationship of cycle, frame and slot. Nodes of different network depth use different time slots. The transmittable time slot for each node depends on its hop distance to the sink according to the following equation, where ℓ_i denotes the hop distance of node i toward the sink.

$$S_i = (\ell_i + 2) \mod 3; \qquad (1)$$

Rule 1: Each source node i sends its own data in the S_ith slot of the ith frame per cycle.

Rule 2: When the data of source node i travels toward the sink, each intermediate node j will forward the data in the S_jth slot of the ith frame.

The transmission scheduling follows the 2 rules above. For example, as shown in Figure 2 (Left), node 3 transmits its own data in slot 1 of the 3th frame and the data is forwarded by node 2 in slot 0 of the 3th frame. That is to say, when a time frame is allocated for a source node, it already includes the bandwidth required for forwarding the data in the network by intermediate nodes. Thus, the slot assignment of RoMac is essentially traffic-driven, significantly departing from traditional assignment schemes that are simply based on topology. In addition, the vertical 2-hop interference is eliminated because node i uses a transmittable slot different from its parent, children, grandparent and grandchildren according to Equation (III-A), and thus does not collide with them. Note that the horizontal 2-hop interference is also eliminated because each node is assigned a unique frame according to its ID.

B. Performance Analysis

This novel frame-slot assignment design not only eliminates 2-hop interference providing collision-free communication, but also possesses several interesting properties.

Constant Jitter and Timeliness. It is easy to see that, given the data stream from a source node, the delay between two successive packets equals to the cycle length and is constant. That is to say, each data flow has constant jitter of $O(n)$, where n is the size of the network. Here we assume all the nodes are periodically reporting data to the sink. We proved in [12] that for many-to-one packet scheduling in wireless networks the time optimum cost is $\max(2n_u - 1, n - 1)$ time slots, assuming each node reports one unit of data in each round. Here n_u denotes the number of sensors in a sink's largest top subtree. A top-subtree is defined as a subtree that has a child of the sink as its root. If the sink receive a packet in every slot, then it takes at least $n - 1$ slots to collect the data from all the nodes in the network for each round. That is to say, RoMac can achieve optimum in terms of packet jitter. RoMac also provides bounded end-to-end delivery latency. When a packet traverses from the source node to the sink, the packet is forwarded in a determined slot of a determined frame by each intermediate node. As shown in Figure 2, the packet sent by node 3 in slot 1 of the 3th frame is forwarded by node 2 and 1 in slot 0 and 2 of the same frame respectively. Note that when a packet traverses 3 hops, it will be forwarded in the same frame of the next cycle, according to Equation III-A. Assume the depth of the source node is d, then the data delivery latency is no more than $d/3$ cycles. We can see that the latency is bounded by $O(d)$.

Resilience to Dynamic Topology. RoMac also achieves resilience to topology dynamics. In RoMac, the transmission schedule of each node i is determined by its frame set F_i and its transmittable slot in each frame S_i. Node i can locally calculate S_i based on Equation III-A using its hop distance toward the sink in the data collection tree. Here we assume each node is aware of its depth in the topology tree, since this information can be easily obtained from most distance vector routing protocols such as MultihopLQI [13] and CTP [14]. Next we show that F_i also can be determined using local information. For each node j in node i's subtree, node i

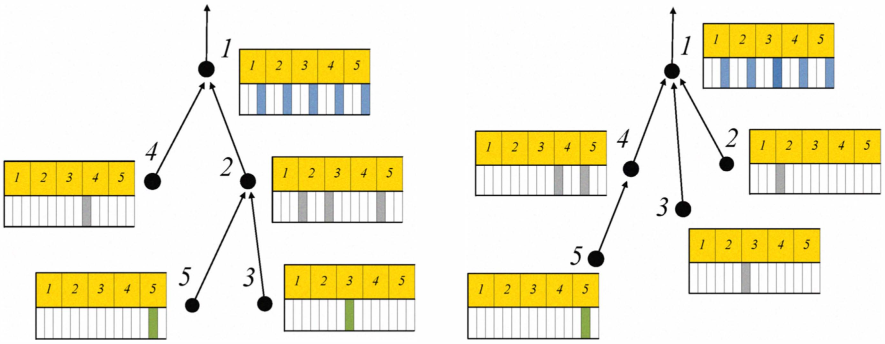

Figure 2. The frame-slot assignment of each cycle for all nodes in a network. For each node, the cell with dark color in the bar denotes its transmittable slot in each frame. (Left) The original schedule for each node; (Right) The nodes update their schedules automatically when node 2 and 3 switch their parents.

forward its data in the S_ith slot of the jth frame. That is to say, the jth frame is in F_i. Thus, F_i can be determined as $\{j|j \in subtree(i)\} \cup i$ where $subtree(i)$ denotes the nodes in the subtree rooted at node i. When the topology changes, node i should adaptively update F_i if certain nodes join/leave node i's subtree. Two approaches can be used for RoMac to update F_i in a fast manner. The first one is utilizing the source address of data packets. If node i receives packets with source address of j, then frame j is in F_i; if no packets come from node j for certain time period, the jth frame should be excluded from F_i. The second approach is to learn from the arriving time of received data packets. If a packet is received by node i during the jth frame, then frame j belongs to F_i. The frame set F_i is updated every TDMA cycle. So when the topology changes, each node i can immediately update its frame set F_i utilizing either of the two approaches and calculate its transmittable slot of each possessed frame using its network depth. For instance, in Figure 2 (Left), both node 3 and 5 choose node 2 as parent, so the transmission frame set of node 2 is $F_2 = \{2, 3, 5\}$; in Figure 2 (right), node 3 and 5 switch their parents to node 1 and 4 respectively. Node 3 changes its transmittable slot to 0 as its network depth changes to 1. Node 3 keeps its transmittable slot since it still has the same hop distance to the sink. The frame-slot assignments of intermediate node 1 and 4 are also updated accordingly. Node 4 uses slot 0 of frame 5 to forward the data packets from source node 5 while node 2 excludes frame 3 and 5 from F_2 since no more packets are received from those nodes. Although the topology changes due to the mobility of node 3 and node 5, each node is able to detect the change and updates its schedule immediately with local information. RoMac is robust in that it is invariant to topology changes.

Ultra-Low Control Overhead. Another advantage of RoMac is that it significantly reduces the overhead of schedule message exchange. When transmission schedules become outdated because of network dynamics, nodes can locally know and update their frame set by monitoring upstream data packets and calculate transmittable slots from the network depth as described above without the need of using explicit scheduling messages.

C. System Design

The basic model of RoMac is simple yet effective. In the design we also consider some possible system dynamics.

1) Initialization With Distributed Ordering: In some sensor network deployments, the node ID is not always in the range of [1, n]. Also, nodes may have different data rates. For example, nodes at hot spots inside a data center may perform sampling at higher rate compared to others. Moreover, the reliability over wireless links is varying. To transmit the same amount of data, nodes with weak link quality shall demand more bandwidth due to link layer retransmissions. So, it is desirable to assign a variable number of time frames in each cycle to data flows based on the traffic demand of each data flow.

Algorithm 1 Distributed Virtual ID Allocation and Ordering

Initially the sink holds the token and initiates the process. The virtual ID starts from 1. Each node u except the sink takes the following actions:

```
if token is passed from parent then
    allocate virtual IDs for $u$ itself;
    initialize all direct children as unmarked;
    while $\exists$ child $i$ unmarked do
        pass the token to child $i$;
        wait for $i$ to return the token;
        mark child $i$
    end while
    return token to parent.
end if
```

RoMac addresses those problems by using a distributed ordering algorithm to assign nodes with virtual IDs. Note that the distributed ordering algorithm only needs to run once at the initialization phase of network deployments. A virtual

ID denotes an assigned frame. The algorithm allocates a range of successive virtual IDs for each source node proportional to their demand at the network initialization phase. For example, if the data rate base is d and node u's data rate is $2d$, then node u is assigned 2 virtual IDs (e.g., $\{k, k+1\}$). The process of ordering is described in Algorithm 1. The virtual ID starts from 1 and increases successively. A network token is used to avoid conflicts during the ordering phase. Only the node granted the token can take the virtual ID allocation action. Initially the sink holds the token and initiates the ordering process. For each node, once receiving the token from its parent, it first allocates a proportional number of virtual IDs for itself. It then passed the token to its children one by one. The node returns the token back to its parent only when all of its children have allocated virtual IDs. Explicit acknowledgment is used to ensure that the token will not be lost. The ordering finishes when the token is finally returned back to the sink, which indicates that all nodes are already assigned a proportional number of virtual IDs. At the same time, the sink also can determine the TDMA cycle size as vid_{max} frames. vid_{max} is largest virtual ID after the allocation action is performed throughout the network. The sink will broadcast messages to notify other nodes of the cycle size, which can be done via reliable data dissemination protocols such as Cascades [15]. It is necessary to lock the network topology during the ordering process, otherwise it may result in inconsistency if some nodes switch parents. Therefore, when nodes discover their paths to the sink, they will temporarily lock their current parents. Once receiving the notification messages, the node will unlock its parent.

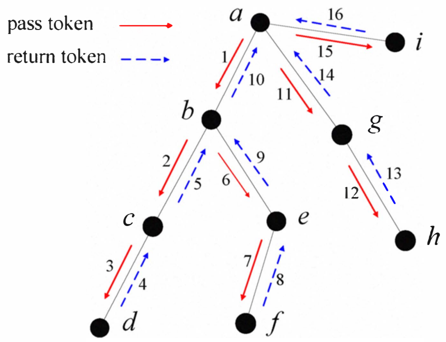

Figure 3. An example of the distributed ordering process. The number on each arrow denotes the action sequence

The distributed ordering algorithm requires much lower message cost than centralized approaches. Here we assume the network size is n. From Figure 3, we can see that the token traverses the topology tree exactly once, therefore the message cost for the distributed ordering process is $O(n)$. For a centralized approach, the sink needs to collect bandwidth demand information and send transmission schedules to all nodes. The worst case is line topology. The overhead is $1 + 2 + \cdots + (n-2) + (n-1) = O(n^2)$. In the best case, where the topology is a balanced tree, the average network hop count is $O(\log n)$, so the message cost is $O(\log n) \times O(n) = O(n \log n)$. We can see that the distributed approach is much more efficient than the centralized one.

IV. ROMAC IMPLEMENTATION IN TINYOS

In this section, we introduce the implementation of RoMac in the TinyOS-1.x [16] operating system, running on the Crossbow iMote2 sensor network platform. The iMote2 sensor mote is an advanced wireless sensor node platform. It is equipped with 256 KB SRAM, 32 MB SDRAM, and an 802.15.4 compliant radio chip CC2420 which provides an effective data rate of 250 kbps. Its PXA271 processor can be configured to work from 13 MHz to 416 MHz. We configured its PXA271 processor to operate in a low voltage (0.85 V) and low frequency (13 MHz) mode in normal operations.

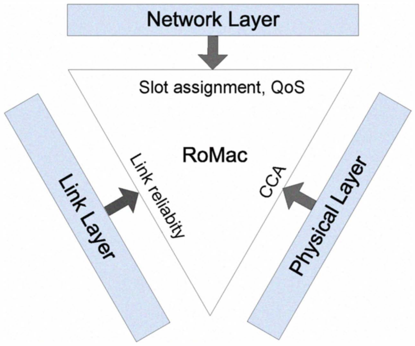

Figure 4. The cross layer design of RoMac

The implementation adopts a cross-layer design to have a joint optimization, as illustrated in Figure 4. It learns its frame assignment from the upstream packets, and the transmittable slots from its network depth in the network layer. The link reliability known from transmission statistics in the link layer helps each node to allocate frames for its own data during the network initialization phase. We activate the CCA (Clear Channel Assessment) function of the CC2420 radio to enhance the protocol robustness and ease the need of fine-grained time synchronization. During the transmittable slots, a node starts transmission only when CCA indicates the channel is idle. The CC2420 radio will backoff for a random period bounded by a time window when it detects the channel busy. RoMac uses a shorter backoff period than the default in CSMA mode since the channel contention is already well mitigated.

Time synchronization has been well studied in the literature making TDMA practical for wireless sensor networks. Hardware-based approaches include using AM receiver add-on [17], and tuning to the magnetic field radiating signal from existing AC power lines [18] to provide global clock source for battery-operated sensor nodes. There are also many software-based in-band time synchronization approaches designed for sensor networks. In our implementation, we adopt FTSP (Flooding Time Synchronization Protocol) [19], a multi-hop time synchronization protocol that achieves high precision performance by utilizing MAC-layer timestamping and comprehensive error compensation, including clock skew estimation. In the original FTSP work [19], timing errors of less than 67 microseconds were reported for an 11-hop network of Mica2 nodes. In our experiments, the timing error is $1 \sim 2$ millisecond. That meets the requirements of RoMac since we enable CCA carrier sensing to ease the need of fine-grained time synchronization.

V. SENSOR TESTBED EVALUATION

A. Experiment Methodology

Table I
EXPERIMENT PARAMETERS

Parameter	Value
Default data transmission power	-25 dBm
Time slot size	10 ms
Schedule update interval (only for TreeMAC)	8 sec
Clock synchronization error (ε_{clock})	$1 \sim 2$ ms
FTSP synchronization message interval	5 sec
Routing beaconing interval	10 sec

To evaluate the performance of RoMac, we conducted experiments on a indoor sensor network testbed comprised of 22 iMote2 motes. While the state-of-the-art data center sensing network deployment [1] used more sensor nodes, the network selected a 30-second sampling rate due to bandwidth constraints. In our experiments, the network samples temperature and energy measurements at a much higher rate of 25 Hz simulating the high fidelity traffic load. Each source node generates a data rate of 3.5 pps (Packet Per Second). The data payload size is 74 bytes. The cumulative topology dynamics observed on the testbed is illustrated in Figure 5. Each source node piggybacks its parent in the upstream data packets. When the *parent* field of a packet received at the gateway is different from a previous one, we count it as a topology change. From Figure 5 we can see that the network is quite dynamic. The data collection routing protocol used is MultihopOasis [13], which is a variant of MultihopLQI in TinyOS-1.x [16] using LQI (link quality indication) as the routing metric. In CC2420 radio, the LQI measurement is a characterization of the strength and/or quality of a received packet. Table I summarizes the key parameters used in the experiments. In following experiments we use 10 ms for each time slot considering the time synchronization errors of FTSP. However, we can use a smaller slot length such as 5 ms when more accurate time synchronization is available using hardware-based techniques [17], [18].

B. Evaluation Results

We compare RoMac to TreeMAC and CSMA following the axis of QoS features and other network performance criteria. The QoS features of RoMac are evaluated under different network configurations by varying the number of source nodes in the network. Note that other evaluation results are based on the setting of a fixed network size of 22 nodes.

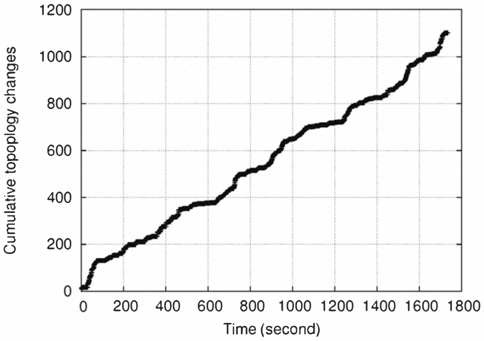

Figure 5. Number of topology changes in the testbed

1) Delivery Latency: We measured the maximum end-to-end delivery latency with network size of $2, 4, 8, 16$, and 22 respectively. As shown in Figure 6, when the number of source nodes is no more than 4, the maximum latency of CSMA is shorter than that of RoMac and TreeMAC. That is because in TDMA a node can transmit only during its scheduled time slots whereas in CSMA, nodes can transmit at any time as long as there is no contention. However, as the number of data flows increases, RoMac obtains better performance than CSMA and TreeMAC with a fast increasing margin. When the number of data source nodes is up to 22, RoMac reduces the maximum delivery delay by 55.1% compared to CSMA. That is because with CSMA the increase in the network size significantly intensifies the channel contention and sensor nodes have to undergo long backoff before the channel is clear for transmissions. For TreeMAC, the transmission schedule can become invalid due to topology dynamics forcing TreeMAC to fall back to the similar performance of CSMA.

2) Packet Jitter: We also measured interval of received packets from all source nodes at the gateway. Figure 7 shows the results under different network sizes. Theoretically RoMac can provide constant jitter assuming reliable links. However, in realistic sensor networks wireless links are characterized as lossy, and link layer retransmissions take place when a packet is not acknowledged. Also, transient and external interference from 802.11 networks also may cause CC2420 to backoff. Therefore, as shown in Figure 7, the jitter of RoMac has certain variation. However, we can see that RoMac can persistently obtain packet jitters close to the optimal value. The theoretical optimum is the

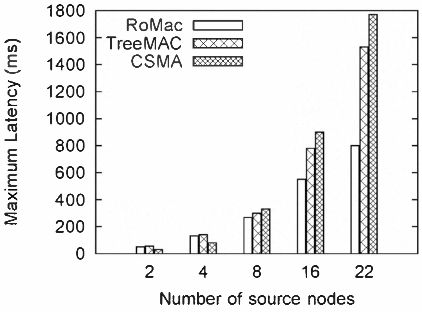

Figure 6. Maximum end-to-end delivery latency v.s. network size

TDMA cycle length under each network configuration. On the contrary, the packet jitter in CSMA varies significantly from 8 milliseconds to 2.5 seconds in our experiments due to its distributed and random backoff nature. TreeMAC provides performance similar to CSMA.

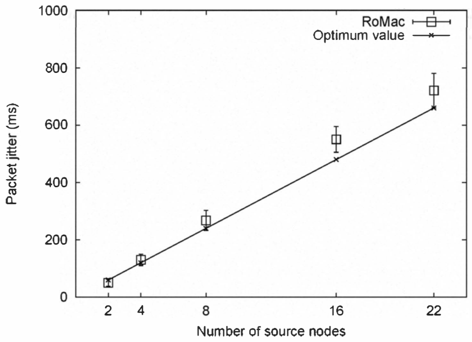

Figure 7. Packet jitter v.s. network size

3) Traffic Adaptability: We show that RoMac is resilient to topology dynamics. In a multi-hop environment, once the network topology changes, the amount of data to be forwarded may vary for nodes in the network, so the assigned bandwidth also shall be adjusted accordingly. Figure 8 shows the demanded bandwidth and the assigned bandwidth of node 1 over running time. The demanded bandwidth is measured as the number of packets transmitted over the radio in each report cycle (5 seconds). The assigned bandwidth is the number of frames a node can use for forwarding data and its own data. Figure 8 shows that the assigned bandwidth is adjusted and thus proportional to the demanded bandwidth when topology dynamics take place. That is to say, RoMac is invariant to topology changes and truly traffic adaptive.

4) Variable Bit-rate Adaptation: In a sensor network the bit rate of a data stream may vary due to the change of QoS or other factors. So the bandwidth allocated for that data flow should also be updated accordingly. In our experiment we use RPC commands to remotely double the sampling rate of sensor node 1. As shown in Figure 9, the number of frames allocated for node 1's data flow changes from 1 to 2, and the number of packets received at the gateway almost increases proportionally.

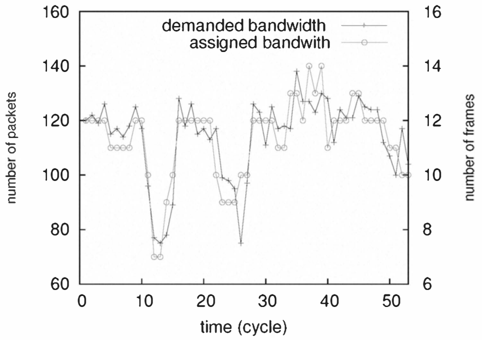

Figure 8. The assigned bandwidth is proportional to the demand

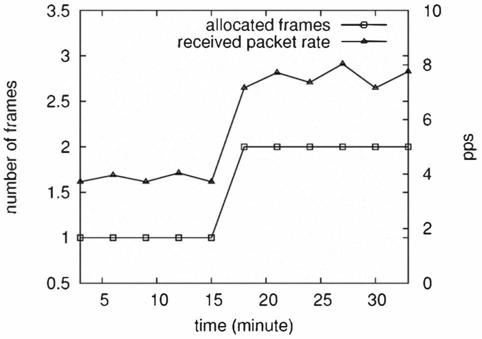

Figure 9. The bandwidth allocated for node 1 increases proportional to its data rate

5) Network Throughput: We measure the network throughput by calculating how many data packets are successfully delivered to the sink from all nodes in one second. The network throughput is averaged every 3 minutes. As shown in Figure 10, RoMac achieves a better throughput than both TreeMAC and CSMA. The throughput gain is between 23.5% and 69.2%. The advantage of RoMac results from its resilience to topology dynamics. Firstly, TreeMAC suffers transmission collisions and radio backoff when topology changes take place, because the transmission schedule may become invalid where two nodes within 2-hop neighborhood may transmit in the same time slot. Secondly, the occurrence topology dynamics can cause mismatch between the bandwidth demand and the assignment of some sensor nodes resulting in congestion. To mitigate the funneling effect in data collection networks TreeMAC assigns bandwidth proportional to the demands of sensor nodes. However, upon route changes there exists a delay for the slot assignment to be updated and disseminated.

6) Network Fairness: Traffic fairness is an important feature that enforces fairness among all data flows. However, high channel contention can cause channel starvation for some nodes in a sensor network. We expect all nodes in a sensor network can fairly deliver their packets to the sink.

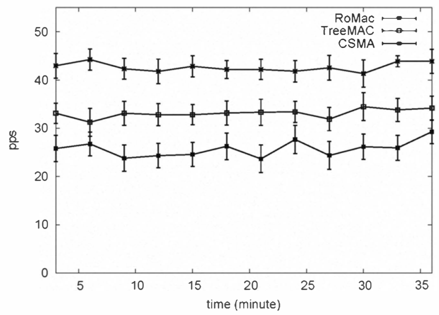

Figure 10. The network throughput over running time

We utilize the definition of fairness index ϕ from [20]: $\phi = \frac{(\sum_{i=1}^{N} r_i)^2}{N \times \sum_{i=1}^{N} {r_i}^2}$. Here, r_i is the average rate of packets delivered from the ith sensor and N is the number of sensors in the network. Figure 11 shows that RoMac offers a much higher degree of fairness than TreeMAC. We can see that the leading margin of fairness index can be up to 40%. The result indicates that RoMac can improve network fairness by regulating channel access. In TreeMAC, the mismatch between the bandwidth demand and the assignment due to topology changes can cause congestion and channel starvation for some nodes.

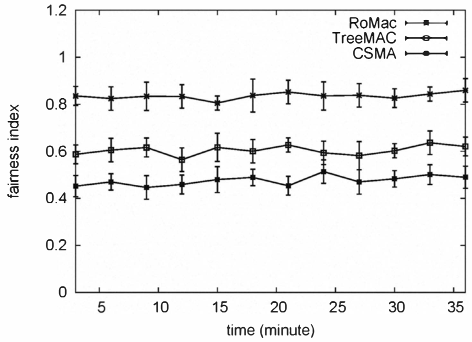

Figure 11. Network fairness over running time

VI. CONCLUSION

In this paper we present RoMac, a localized Qos-aware TDMA MAC protocol for high-fidelity data center sensing networks. Besides collision-free scheduling, it can achieve resilience to the dynamics in low-power wireless communication and provision persistent QoS guarantee in terms of data reliability and real-time data delivery. Real-time data acquisition and control is also critical for a variety of cyber physical systems. Our generic protocol can well apply to those applications.

REFERENCES

[1] C.-J. M. Liang, J. Liu, L. Luo, A. Terzis, and F. Zhao, "RACNet: A High-Fidelity Data Center Sensing Network," in *SenSys*, November 2009.

[2] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-mac: a hybrid mac for wireless sensor networks," in *SenSys*, 2005.

[3] G.-S. Ahn, E. Miluzzo, A. T. Campbell, S. G. Hong, and F. Cuomo, "Funneling-mac: A localized, sink-oriented mac for boosting fidelity in sensor networks," in *SenSys*, 2006.

[4] W.-Z. Song, R. Huang, B. Shirazi, and R. Lahusen, "TreeMAC: Localized tdma mac protocol for high-throughput and fairness in sensor networks," in *PerCom*, March 2009.

[5] O. Gnawali, R. Fonseca, K. Jamieson, and P. Levis, "CTP: Robust and efficient collection through control and data plane integration," February 2008.

[6] W. Ye, J. Heidemann, and D. Es, "An energy-efficient mac protocol for wireless sensor networks," in *INFOCOM*, 2002.

[7] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *SenSys*, 2003.

[8] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle mac with scheduled channel polling," in *SenSys*, 2006.

[9] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *SenSys*, 2004.

[10] M. Buettner, G. Yee, E. Anderson, and R. Han, "X-mac: A short preamble mac protocol for duty-cycled wireless sensor networks," in *SenSys*, Boulder, Colorado, USA, November 2006.

[11] Y. Sun, O. Gurewitz, and D. B. Johnson, "Ri-mac: A receiver initiated asynchronous duty cycle mac protocol for dynamic traffic load." in *SenSys*, November 2008.

[12] W.-Z. Song, F. Yuan, and R. Lahusen, "Time-optimum packet scheduling for many-to-one routing in wireless sensor networks," in *IEEE MASS*, 2006.

[13] W.-Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, and R. Lahusen, "Air-dropped sensor network for real-time high-fidelity volcano monitoring," in *MobiSys*, June 2009.

[14] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *SenSys*, 2009.

[15] Y. Peng, W. Song, R. Huang, M. Xu, and B. Shirazi, "Cacades: a reliable dissemination protocol for data collection sensor network," in *IEEE Aerospace Conference 2009*, March 2009.

[16] "TinyOS: http://www.tinyos.net/tinyos-1.x."

[17] R. Mangharam, A. Rowe, R. Rajkumar, and R. Suzuki, "Voice over sensor networks," in *RTSS*, December 2006.

[18] A. Rowe, V. Gupta, and R. Rajkumar, "Low-power clock synchronization using electromagnetic energy radiating from ac power lines," in *SenSys*, November 2009.

[19] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi, "The flooding time synchronization protocol," in *SenSys*, November 2004.

[20] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating congestion in wireless sensor networks," in *SenSys*, November 2004.