Mode Switch - Adaptive Use of Delay-Sensitive Or Energy-Aware Communication in IEEE 802.15.4-Based Networks

Feng Chen*‡, Xiaoyu Tong†, Edith Ngai† and Falko Dressler*
*Computer Networks and Communication Systems, University of Erlangen, Germany
†Department of Information Technology, Uppsala University, Sweden
‡Siemens AG, Industry Automation Division, Germany

***Abstract*—We propose "mode switch", an adaptive load-sensitive solution that supports both an energy-efficient operation mode for transmitting normal sensor data and an QoS-aware low-latency mode for high priority emergency messages. Typically, sensor networks are considered to operate efficiently w.r.t. energy consumption. A good example is the IEEE 802.15.4 protocol, which has mainly been designed for this purpose. The standard IEEE 802.15.4 beacon-enabled cluster-tree topology supports energy-efficient operation for low-rate data transmissions. However, in the case of detected alarms, high-priority emergency messages need to be transmitted with low-latency and guaranteed delivery. In this case, the cluster-tree topology appears to be inefficient and cannot provide the needed support. In this paper, we show how energy-efficiency and delay-sensitivity can be combined by developing an adaptive solution that completely switches between the fundamentally different operation modes beacon-enabled cluster-tree mode and non-beacon-enabled mesh mode. The key challenge is to provide a fast mode switch that also keeps track of necessary topology information. We present a protocol for this mode switch and demonstrate, based on extensive simulations, that the performance demands can be fulfilled with only reasonable overhead even for mid-size networks. Our simulation results clearly indicate that the system achieves the objected QoS and energy-efficiency with "mode switch" in real-time.**

I. Introduction

The IEEE 802.15.4 cluster-tree networking can be extremely energy-efficient thanks to its distributed beacon-enabled synchronization scheme allowing low-duty-cycled operation at each device [1], [2]. In addition, the cluster-tree structure has integrated routing capabilities, which minimize the overhead for route maintenance. Therefore, it is quite suitable for many sensor network applications requiring multi-hop communications and stressing long network lifetime [3]. However, this protocol has very limited real-time capabilities due to its low-duty-cycled operation, which restricts the available bandwidth in the network. Furthermore, the structured cluster-tree routing scheme prevents the development of any of other QoS-aware routing schemes. In contrast to the cluster-tree topology, unstructured IEEE 802.15.4 mesh networks provide much better performance, especially w.r.t. low-latency transmissions. It operates in full-active mode and may use specific latency-aware routing algorithms.

Many practical applications, however, require both energy-efficiency and real-time capabilities. Studying such applications in detail, the networks usually need to operate in a low-power mode most of time, i.e. routine sensor data is transmitted at a low data-rate to a sink. Without loss of generality, it can be assumed that such a routine reporting is not delay-sensitive. Envisioning more intelligent sensor nodes able to pre-process collected sensor information, we can further assume that these sensor nodes will be able to determine unusual situations, e.g. because measures exceed predefined thresholds. Due to the significance of the unusual events, sensors are usually configured to sample the readings at a higher frequency, thus produce a burst of data at a high rate. Furthermore, this data often requires to be transmitted to the sink in real-time, especially when some measures need to be taken to handle the emergent events. Therefore, the sensor network is expected to have sufficient bandwidth to cope with the suddenly increased traffic and to provide real-time transmissions. However, the sensor network, which is configured for energy-efficient operation, lacks of such capabilities.

In this paper, we propose "mode switch", an adaptive load-sensitive solution that supports both an energy-efficient operation mode for transmitting normal sensor data and an QoS-aware low-latency mode for high priority emergency data (Sections III and IV). It supports real-time deconstruction and reconstruction of cluster-trees for switching between the two operation modes in IEEE 802.15.4 based sensor networks. We carefully evaluated our protocol in extensive simulations (Section V). According to the achieved performance results, the proposed protocol shows all the intended characteristics: energy-efficient operation in the normal case and low-latency communication for emergency messages. The switching can be performed in a timely manner with low overheads even for mid-size networks.

II. Related Work

In this section, we briefly introduce IEEE 802.15.4 and discuss related approaches for operating wireless and sensor networks in either energy-efficient or QoS-aware modes.

A. IEEE 802.15.4

IEEE 802.15.4 [1], [2] is a standard that specifies the physical layer and media access control designed for low-

978-1-4244-7489-9/10/$26.00 ©2010 IEEE

rate wireless personal area networks (LR-WPANs). IEEE 802.15.4 MAC includes features such as its dual operational modes (non-beacon-enabled mode/beacon-enabled mode). Its beacon-enabled mode can conserve energy by using the RF sleep mechanism, but it is limited by the lower data throughput. On the other hand, the non-beacon-enabled mode can offer higher data throughput, which makes this standard more attractive for providing multimedia services over the networked sensors, but at the expense of significant energy consumption [4].

The standard supports two topologies: a star and a peer-to-peer topology. In star networks, the communication occurs only between a central PAN coordinator and the sensor nodes. Both beacon-enabled and non beacon-enabled MAC operation modes are supported in the star topology. Furthermore, star networks can optionally use the Guaranteed Time Slot (GTS) mechanism, which provides contention-free access for guaranteed-latency transmissions. The peer-to-peer topology allows more complex mesh networking. Thus, it has the advantage of covering larger areas and a higher scalability. However, an unstructured IEEE 802.15.4 mesh network can operate only in the non-beacon-enabled mode. It inherently lacks the ability of energy-efficient operation if low-power operation is desired.

A special case of the peer-to-peer topology is the cluster-tree topology, which also allows the beacon-enabled operation in MAC layer. The standard defines a distributed beaconing scheme to form a cluster-tree PAN and to maintain synchronization between parent nodes and and their children. In order to avoid beacon collisions among all beaconing devices, there are generally two rules for deploying a beacon scheduling scheme: First, as illustrated in Figure 1, a coordinator that is not the PAN coordinator should guarantee that its outgoing superframe does not overlap with its incoming superframe, which is defined by its father. Secondly, the outgoing superframes of any two beaconing devices, even of those having no father-child relationship, must not overlap (at least as long as those nodes are in a common communication range). This rule implies that in the worst case the timing of all outgoing superframes in the whole network must be accommodated within the length of one Beacon Interval (BI) without overlapping. In addition, the standard requires that Beacon Order (BO) and Superframe Order (SO), which are used to control the length of the beacon interval and that of the active part, shall be equal for all superframes on a PAN. One consequence of these design rules is the need of choosing a very low duty-cycle, especially in a large dense cluster-tree network.

B. Performance Improvements

Some analysis of the performance of IEEE 802.15.4 networks has been conducted in [5], [6]. Kim et al. proposed a priority-based scheme comprising Frame Tailoring (FRT) and Priority Toning (PRT) to reduce latency in event-

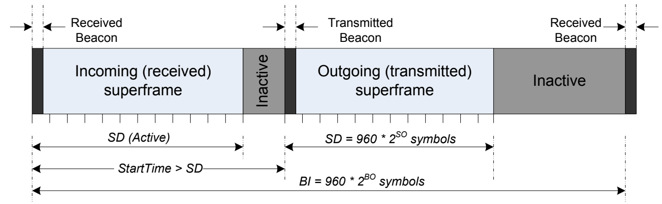


Figure 1. Superframe structure with incoming and outgoing superframes

monitoring IEEE 802.15.4 networks [7]. Their methods are contention-based and cannot provide guaranteed latency bounds. Koubaa et al. analyzed the delay bounds guaranteed by the IEEE 802.15.4 GTS allocations for real-time sensor networks using the analytical Network Calculus formalism [8]. Based on the analysis, they further pointed out the limitations of the explicit GTS allocation in IEEE 802.15.4 and proposed an implicit GTS Allocation Mechanism (i-GAME) [9]. Their new approach improves the bandwidth utilization of the original GTS mechanism, but at the cost of increasing guaranteed delay bounds which is not applicable for applications with very strict real-time requirements. Although the above works intend to provide certain level of QoS with IEEE 802.15.4, they focus on the MAC layer design rather than a joint optimization with the network layer protocol. Also, they do not consider the energy consumption and its trade-off with the expected delay. In our previous study, we have investigated the IEEE 802.15.4 star networks for industrial factory automation, in which strict real-time bounds can be guaranteed.

Energy-efficient cross-layer design has been widely studied for wireless networks [10]. Cho et al. [11] proposed a variable bandwidth allocation scheme to reduce the energy consumption for sensor networks, which have large spatial variation in the sensor distribution. Xiao et al. [12] proposed a power scheduling scheme which suggests that the sensors with bad channels or poor observation qualities should decrease their quantization resolutions or simply become inactive in order to save power. Kozat et al. considered the problem of energy-efficient communication in wireless multi-hop networks with the objective of providing the end-to-end QoS guarantees to a set of sessions. They addressed a joint problem of power control and scheduling with the objective of minimizing the total transmit power subject to the end-to-end QoS guarantees for sessions in terms of their bandwidth and bit error rate guarantees. However, the above work focus mainly on achieving the optimal scheduling scheme in the MAC layer and its duty cycles, while our work considers routing and MAC protocols coherently and provides energy-efficient QoS routing adapts to the network traffic.

Similarly, QoS routing has been widely studied in wireless networks. He et al. [13] propose SPEED, a protocol which combines feedback control and non-deterministic quality of service aware geographic forwarding. Lu et al. [14] describe a packet scheduling policy, called velocity monotonic scheduling, which inherently accounts for both time and distance constraints. Felemban et al. [15] have proposed the Multi-path and Multi-Speed Routing Protocol (MMSPEED) to provide probabilistic QoS guarantees in sensor networks with multiple QoS levels related to the timeliness of data delivery. They suggest using different delivery speeds and probabilistic multipath forwarding in the reliability domain. Ergen et al. [16] present a routing algorithm that maximizes the sensor network lifetime. It furthermore incorporates delay guarantees into energy efficient routing by limiting the length of paths from each sensor to the collection node. Random Re-Routing (RRR) is an adaptive randomized routing algorithm, which can relieve network congestion and provide QoS in data delivery [17]. It is designed for generic sensor networks which require fast report after occurrence of unusual events [18]. Most of these works focus on the design in the network layer without considering the specific characteristics of MAC layer protocols. Although there is some work on jointly providing delay and energy efficiency [19]–[22], none of them suggests the possibility of switching routing policy and the operation mode in MAC layer cooperatively at the same time for better network performance. In this work, we propose "mode switch", which is an adaptive and load-sensitive solution that supports energy-efficient and QoS communication in different modes according to the change of the environment.

III. Target Applications and Requirements

Our proposed mode switch scheme aims at providing IEEE 802.15.4 based sensor network solutions for industrial applications that require multi-hop and energy-efficient wireless networking. The structure of a typical topology is shown in Figure 2, where a group of IEEE 802.15.4-compliant sensor nodes (numbered from 26 to 49) are scattered within a target area to perform monitoring tasks. These sensor nodes are connected to a tree-based multi-hop network that transmits the collected data to a central gateway, which represents the PAN coordinator in the cluster-tree structure.

In normal conditions, sensor nodes perform sensing at constant intervals and generate a steady volume of data at a low rate. We call this *routine data*. There are no strict real-time requirements for transmitting this data (routine reporting). However, there are energy constraints requiring the network to operate in low-power mode to prolong the network lifetime [3]. When emergent events of special interest occur, a large amount of sensor data may need to be generated to clearly identify and to monitor the event, which finally causes an increase of network traffic. We call this *emergent data* in the scope of this paper. For example, if some the temperatures at the six red-colored nodes in Figure 2 increases dramatically, the sensor nodes collect samples at a high rate producing a burst of data. We assume that sensor nodes can distinguish routine data and emergent data by using a similar method of detecting the content change in sensor measurements [18]. In contrast to routine packets, emergent packets are much more delay-sensitive and require more bandwidth. It is also assumed that emergent events occur infrequently and each single one may not last long.

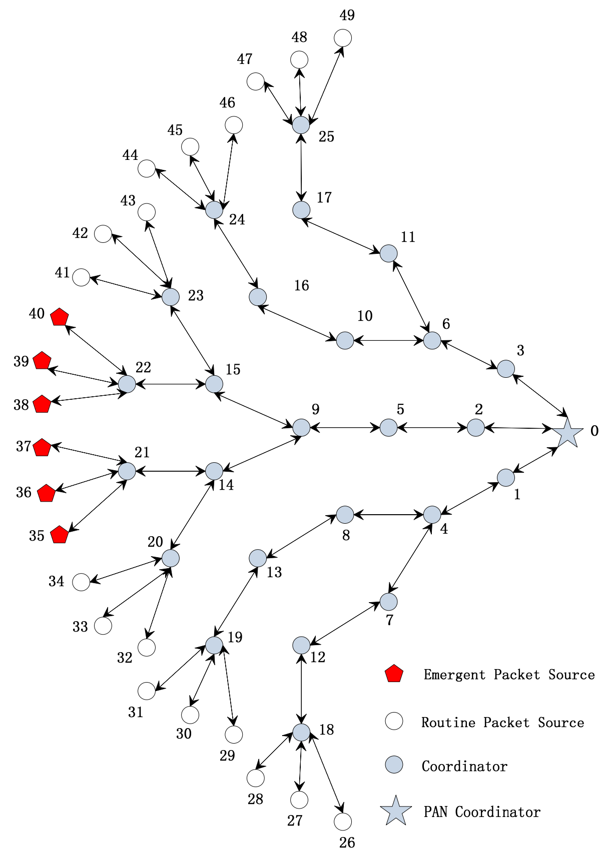


Figure 2. A sample of the 802.15.4 cluster-tree networks

We support such application demands with our "mode switch" solution. Basically, the developed system is able to adaptively switch between energy-efficient cluster-tree networking in the beacon-enabled mode and low-latency shortest path routing in mesh mode without beaconing. The challenging part is the low-overhead detection of the current situation and the fast and, again, low-overhead switch between the operation modes.

IV. Mode Switch Algorithm

We propose a mode switch scheme that allows IEEE 802.15.4 based sensor networks to switch between cluster-tree mode and mesh mode adaptively according to the desired performance. In normal conditions, the network performs routine reporting using the energy-efficient cluster-tree routing. Upon detection of the increased traffic caused by the occurrence of emergent events, the network temporarily de-constructs the cluster-tree to operate in a fully active mesh

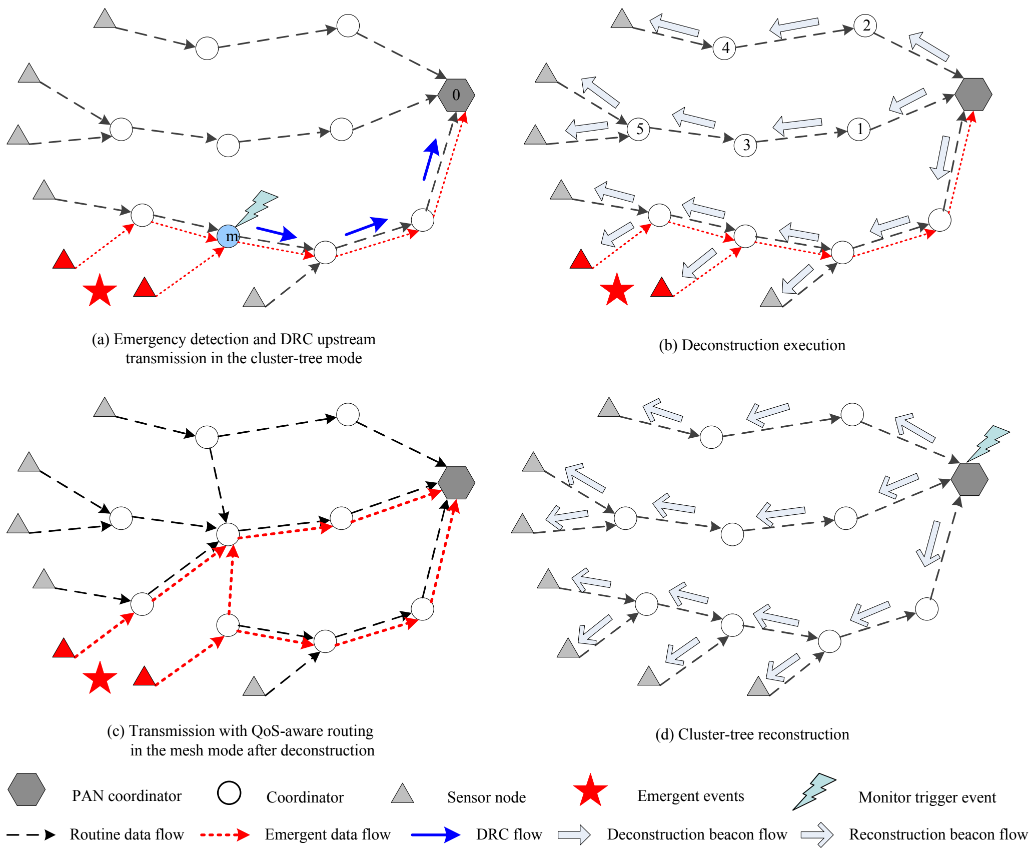


Figure 3. De-constructing and re-constructing cluster-tree triggered by traffic monitoring of the emergent events

mode in which certain of QoS-aware routing schemes can be applied to provide desired QoS for emergency reporting.

After the emergent event ends, the network can switch back to the low-power mode by resuming the cluster-tree structure according to the previously stored network settings. The work flow of the switch scheme is depicted in Figure 4.

In the following, we describe the different parts of the algorithm in more detail, focusing on the cluster-tree deconstruction and the necessary steps to re-construct it after leaving the mesh mode.

A. Cluster-tree deconstruction

Initially, the network starts with a cluster-tree topology and forwards packets from the source to the sink along

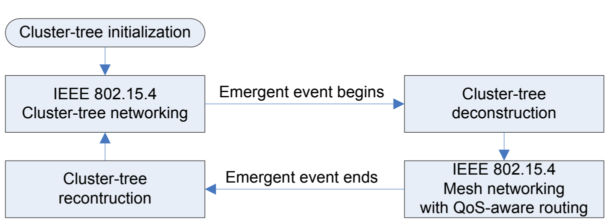


Figure 4. The work flow of the mode switch algorithm

fixed cluster-tree routing paths. To detect the trigger event for deconstruction, each coordinator keeps monitoring the incoming rate of the emergent packets τ_d. The monitoring interval is set equal to a length of BI (Beacon Interval). When the first coordinator in the network detects that the observed τ_d exceeds a threshold θ_D, the deconstruction process will be initiated, consisting of the following two stages.

1) Stage 1: Upstream reporting of deconstruction request command: This stage is illustrated in Figure 3(a). One coordinator that detects the trigger event will generate a Deconstruction Request Command (DRC) at the network layer and send it to the PAN coordinator using the Contention Access Period (CAP) in IEEE 802.15.4. If a set of coordinators detects the increased traffic, more than one DRC will be generated and transmitted simultaneously through the network. Deploying such a redundant mechanism can increase the probability of success for transmissions in a contention-based manner to ensure that at least one DRC can reach the PAN coordinator. However, too many DRC transmissions may increase the traffic load significantly, resulting in QoS degradation for data packets. In order to balance the degree of redundancy and the probability of success, we developed a DRC transmission algorithm. The algorithm is depicted in Figure 5 and discussed in the following.

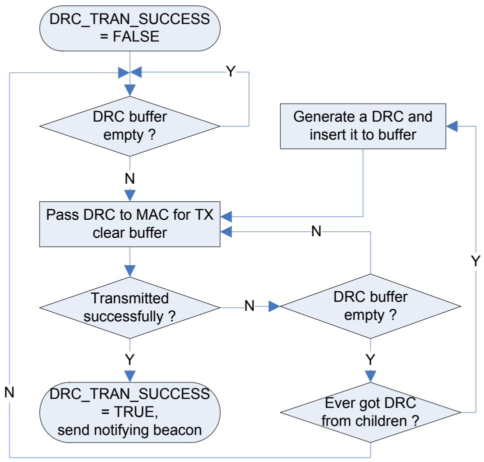


Figure 5. Deconstruction request command transmission algorithm

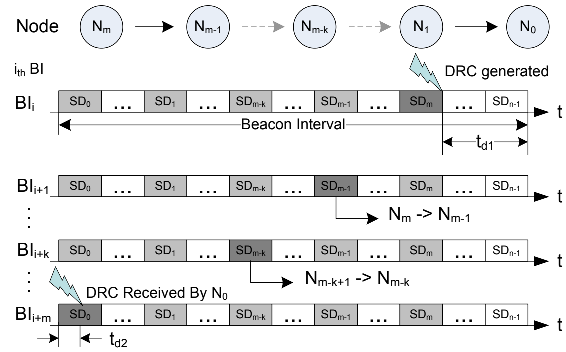


Figure 6. Delay analysis for DRC reporting

Each coordinator has a buffer at the network layer to store at most one DRC. Every time when a DRC is either generated by its own deconstruction monitor or received from its child coordinators, it will be buffered if the buffer is empty. Furthermore, the variable CMD_TRANS_SUCCESS is used to control that at most one DRC is transmitted per switching process. Initially, it is set to FALSE.

When the buffer is filled for the first time, the buffered DRC will be passed to the MAC for transmission. Since the DRC is expected to be transmitted as quickly as possible during its upstream forwarding, a priority queue is used between the network and the MAC layer to give highest priority to the transmission of the DRC. If the MAC reports successful transmission of the DRC, CMD_TRANS_SUCCESS is set to TRUE. Meanwhile, the coordinator sets a flag in its next outgoing beacon to tell all its child coordinators not to transmit any further DRC. If the transmission fails, another DRC retransmission is scheduled at the network layer.

Once CMD_TRANS_SUCCESS is set TRUE due to a successful DRC transmission, or due to the notification from a father node in the tree, this coordinator should not initiate any DRC transmissions in current switch process, even when the buffer is filled again.

2) Estimation of DRC reporting delay: We conducted an analytical calculation to estimate the reporting delay d_{DRC} of the DRC, which is defined as the time from the generation of a DRC to its arrival at the PAN coordinator. We consider a general case of the DRC transmission along a branch in a cluster-tree network, where there are a total of n beaconing devices. Let the DRC be generated at the coordinator N_m, which is m hops away from the PAN coordinator N_0. The beaconing sequence, i.e. the order of outgoing superframes, within a BI for all the coordinators along the forwarding path from N_m to N_0 is depicted in Figure 6. More specifically, N_{m-k} receives packets from its child coordinator N_{m-k+1} in its outgoing superframe denoted by SD_{m-k} and forwards them to its father coordinator N_{m-k-1} in SD_{m-k-1}. Note that the outgoing superframes of two father-child coordinators can be not adjacent to each other, depending on the specific beacon scheduling scheme.

According to our design, N_m will generate a DRC at the end of SD_m within the i_{th} BI denoted by BI_i, after it detects the number of received emergent packets in current outgoing superframe exceeding the threshold θ_D. The actual begin of the DRC transmission will be deferred to SD_{m-1} in BI_{i+1}. Since a priority queue is deployed to transmit the DRC with highest priority at the MAC, the data traffic will have minor impact on the DRC transmission. We assume in our calculation that the DRC can always be successfully transmitted by a coordinator within its next outgoing superframe. Thus, we can infer that the DRC will arrive at the PAN coordinator in BI_{i+m}, which is illustrated in Figure 6. The DRC reporting delay can be calculated as:

$$d_{DRC} = t_{d1} + (m-1) \times l_{BI} + t_{d2} \qquad (1)$$

Here, l_{BI} is the length of a BI. t_{d1} denotes the delay from the generation of the DRC to the beginning of the BI_{i+1}, which is smaller than l_{BI}. The value of t_{d1} is determined by the specific beacon scheduling scheme and is bounded by $l_{BI} - (m+1) \times l_{SD}$, where l_{SD} is the length of a Superframe Duration (SD). t_{d2} denotes the transmission delay in BI_{i+m} and is bounded by l_{SD}.

The boundary of d_{DRC} under the above assumptions can be estimated as:

$$\begin{aligned} d_{DRC} &< l_{BI} - (m+1) \times l_{SD} + (m-1) \times l_{BI} + l_{SD} \\ d_{DRC} &< m \times (l_{BI} - l_{SD}) \qquad (2) \end{aligned}$$

3) Stage 2: downstream execution of deconstruction: The actual operations of the cluster-tree deconstruction are executed using the periodic downstream beacon flow as shown in Figure 3(b). Upon receiving the first DRC, the PAN coordinator will set a deconstruction request bit in its next beacon, which we call Deconstruction Beacon (DB). After sending out the DB, the PAN coordinator switches to the beacon-disabled mode by setting the BO to 15 at the

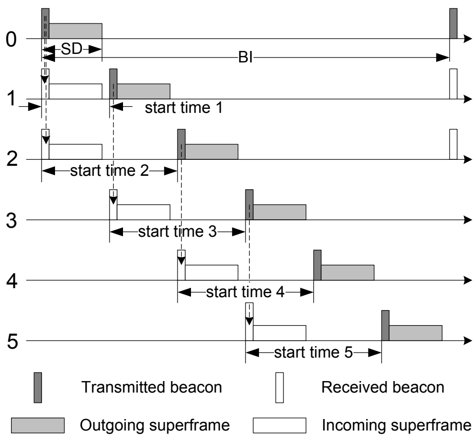


Figure 7. A sample of beacon scheduling

MAC and stops transmitting periodic beacons. After receiving the DB from their father coordinators, all other coordinators re-transmit a DB to their children nodes and then perform switch operation. The deconstruction process ends when all the nodes entered the non-beacon mode.

We now analyze the time needed for the execution of deconstruction T_{d_s2} using an example of a small cluster-tree network, which is a branch of the one in Figure 3(a) considering only the numbered nodes. Except for the PAN coordinator (node 0), all other nodes are assumed to be coordinators and each may be associated with a set of sensor nodes. Without loss of generality, we consider the worst case scenario, in which any two subtrees are within interfering range. Figure 7 shows an example of a collision-free beacon scheduling that has evenly distributed all non-overlapping outgoing superframes within a BI. If no beacon collision occurs in the execution stage, T_{d_s2} will be a constant value, which is smaller than one BI.

However, beacon collision may occur if all coordinators enter the non-beacon mode immediately after sending out the DB. For example, according to the schedule in Figure 7, node 1 transmits DB and enters the non-beacon mode earlier than node 2. The unscheduled transmissions in the full-active mesh mode in subtree 1 may collide with the DB transmission at node 2. As a consequence, if node 4 misses the DB, the whole subtree rooted at node 4 will not be de-constructed, because the DB is transmitted only once at each coordinator.

According to this analysis, we must ensure that all nodes in the network enter the non-beacon mode after the DB has been spread through the entire network. Since T_{d_s2} is bounded by a BI, mode switch can be executed simultaneously by all the nodes a delay of one BI after the PAN coordinator transmits the first DB. However, in standard IEEE 802.15.4 cluster-tree networks, each beacon receiving node is only aware of the time when its father coordinator transmits beacons. Our solution is to let DB carry an accumulated time value in its payload field, to which each coordinator that receives the DB adds its *StartTime* value and then attaches the updated value to its outgoing DB. Upon reception of the DB, each node can derive the latest beaconing time at the PAN coordinator and schedule the delayed operation of the mode switch. For example in Figure 7, node 5 will receive a DB containing the *StartTime* values of nodes 1 and 3, based on which the DB transmitting time at node 0 can be easily calculated.

B. Cluster-tree reconstruction

After the complete deconstruction of the cluster-tree, the network is operating in the beacon-disabled mesh mode with full duty-cycle as shown in Figure 3(c). To detect the finish of the emergent events, the PAN coordinator runs a reconstruction monitor to observe the incoming rate of the emergent packets τ_r. When the observed τ_r falls below a threshold θ_R for n consecutive observations, the PAN coordinator can initiate the cluster-tree reconstruction process.

The reconstruction monitor must be activated by the PAN coordinator a short period of time T_{rm} after it stops beaconing during the deconstruction process. The reason for introducing such a delay is to avoid that the immediate reconstruction is wrongly triggered by the PAN coordinator, which may observe a low τ_r while the network remains in a partially de-constructed state before complete deconstruction. According to our experiments, it is sufficient to set T_{rm} with a length of BI, with which the network can finish executing deconstruction.

Instead of forming a completely new cluster-tree topology, which is an expensive and time-consuming process, the network only needs to recover the previous settings. For example, the association relationship among all the nodes and the beacon scheduling will remain unchanged. To achieve rapid reconstruction, the values of some critical parameters including the association relationship, BO, SO, and the *StartTime* must be stored at each beaconing node before deconstruction. The reconstruction process is initiated by the PAN coordinator simply through restarting beaconing. After rehearing the beacon from its father coordinator, each child coordinator first follows the schedule in the incoming beacon, and then restarts transmitting beacons to its child nodes. When beacons spread through the whole network, the reconstruction process is complete. This process is outlined in Figure 3(d).

Note that during the reconstruction beacon may collide, when their neighboring nodes are transmitting arbitrary data in the beacon-disabled mesh mode. Without causing malfunction of network, the potential occurrence of beacon collision will only prolong the reconstruction process, the degree of which depends on the level of the routine traffic.

V. Simulation Results

We have implemented the mode switch algorithm as an extension to our IEEE 802.15.4 model, which was developed using the OMNeT++ network simulator. It includes a highly detailed models of the standard protocols as well as a battery model that supports energy measurement. The original model has been well validated in detailed studies of the IEEE 802.15.4 star networks w.r.t. their comprehensive performance and real-time capabilities in industrial environments [23].

A. Common simulation settings

To evaluate the mode switch algorithm, we have focused on the sensor network application described in III. The most important simulation parameters are listed in Table I. At the start of the simulation, 50 nodes, including one PAN coordinator, 25 coordinators and 24 sensor nodes, form the cluster-tree topology as shown in Figure 2. The network is configured to use a similar beacon schedule scheme as shown in Figure 7, which evenly distributes all outgoing superframes in a globally non-overlapping fashion within a BI. All sensor nodes except the ones marked with red color start to generate routine packets 10 s after the simulation starts. The routine packets are generated by each sensor node at a constant interval with $\frac{1}{\lambda_R} = 3$ s and transmitted using the cluster-tree routing (CTR). CTR operates in the beacon-enabled mode in which the cluster-tree PAN maintains synchronization between the parent nodes and their children. Scheduling scheme is applied with low duty-cycle to avoid collisions. However, data may not be forwarded immediately until the next allocated outgoing superframe. When the emergent event occurs, six red colored sensor nodes produce emergent packets at a packet rate of λ_E, which follows an exponential distribution. The data payload of all packets is fixed to 10 Byte.

According to our design, the network switches to the mesh mode in order to achieve lower latency for emergency data. Various QoS routing protocols can be applied in the mesh mode. In our simulations, we have implemented a shortest-path geographic routing algorithm (SPR), with which each node forwards packets with best-effort to its neighbor closest to the destination. The routing table of the SPR will be built during the network initialization phase. We have also implemented a SPR algorithm with priority queues to provide differentiated QoS for routine data and emergency data.

Table I
SIMULATION PARAMETERS

Parameter	Value
Network area	800 m × 1000 m
Number of coordinators / sensor nodes	25, 24
Number of routine / emergency sources	18, 6
Simulation time	1500 s
Emergent event start, end time	1000 s, 1300 s
Routine packet rate	λ_R
Emergent packet rate	λ_E
Deconstruction, reconstruction threshold	θ_D, θ_R
Data packet payload	10 Byte
Priority queue size	10 for each class

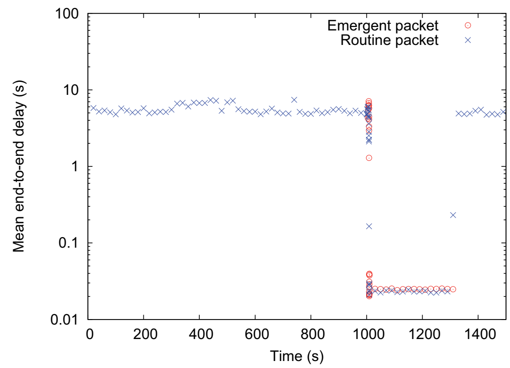


Figure 8. End-to-end delays with switch scheme

B. Effects of the deconstruction scheme

In the first set of experiments, we ran the simulation for 1500 s, which includes exactly one deconstruction and one reconstruction process. The emergent event occurs at $t_e = 1000$ s and lasts for 300 s. We set the (BO,SO) to (6,1), which corresponds to a BI of 0.983 s and a SD of 0.03 s. The deconstruction monitor threshold θ_D and the reconstruction monitor threshold θ_R were set to 5 and 0.5 respectively.

The mean end-to-end delays from one simulation run with $\lambda_E = 5$ pkt/s are plotted in Figure 8. The measuring interval was set to 20 s in the steady state and was shorten to 0.1 s during the switching period. When running in the cluster-tree mode before $t_e = 1000$ s, the network transmits routine packets using the CTR with delays of more than 4 s due to the low duty-cycle operation at MAC. When the deconstruction process is triggered, the network switches to non-beacon mode and transmits packets using SPR. It can be clearly observed that the delays in the mesh mode drop significantly after the deconstruction. When the emergent event ends at 1300 s, the network resumes CTR.

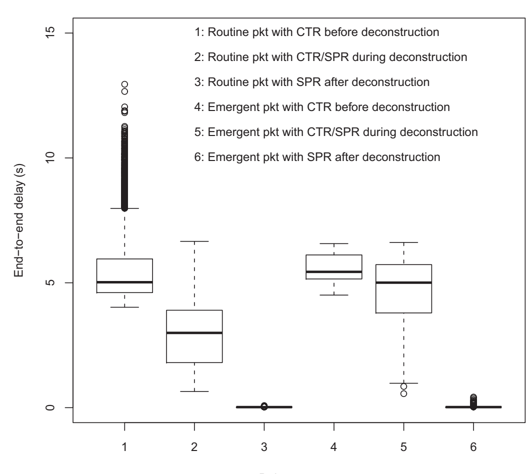
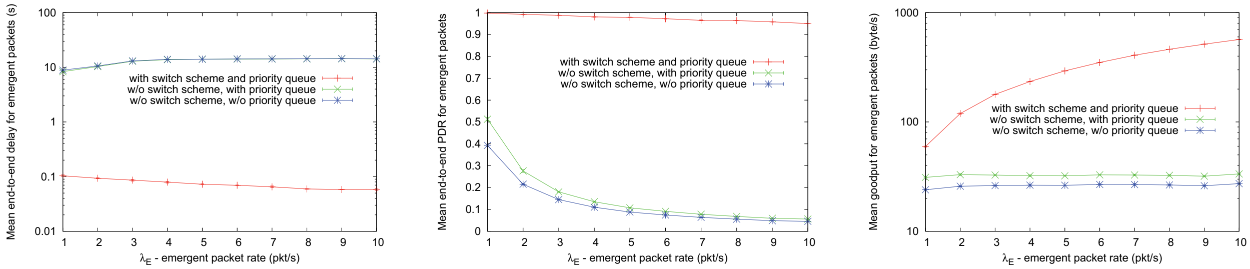


Figure 9. Delay statistics in six categories with switch scheme



(a) Mean end-to-end delay (b) Mean end-to-end PDR (c) Mean end-to-end goodput

Figure 11. Effect of switch scheme on performance of emergency reporting

For the same set of experiments, Figure 9 shows the statistics of the delays in form of box plots. To reveal the transition behavior of the deconstruction process, we have classified all the packets into six categories: categories 1-3 group the routine data while categories 4-5 contain the emergency packets. The subcategories represent the three phases: CTR, deconstruction phase, and SPR. As can be seen, the median value of the category 4 is higher than the one of the category 1, because the emergent packets generated at a much higher rate than the routine packets significantly increase the traffic load. The delays experienced during the deconstruction phase show a much higher variance, because the change of routing policy for each packet may occur at different level in the cluster-tree while the network executes deconstruction in a top-down manner.

Figure 10 illustrates the energy consumption of Node 9, which is an intermediate node in the topology. The amount of energy consumed has been measured within a short periods of 10 s. The much lower energy consumption in the cluster-tree mode meets exactly the requirements of typical sensor network applications that spend most of time on low-rate routine reporting. Relative high power in the mesh mode is traded for higher bandwidth and significantly improved QoS.

In Figure 11, we compare the adaptive switch scheme with two cluster-tree solutions with or without priority queuing that provides differentiated service for routine and emergent packets at MAC. The performance of the emergency reporting is evaluated in terms of delay, packet delivery ratio (PDR), and goodput for different λ_E. We can observe that the solution of priority queuing without the switch scheme slightly improves the PDR and goodput, especially when λ_E is close to λ_R. However, using the CTR, it is not possible to provide satisfying QoS for emergency messages, which have been generated at a high rate and cause a network congestion in the cluster-tree network.

In summary, our mode switch scheme can adaptively change the operating mode to meet both energy and real-time requirements. To support a long network lifetime for routine sensing, the IEEE 802.15.4 network works in the cluster-tree mode with a low-power as depicted in Figure 10. When infrequent emergent event occurs, the network temporarily switches to the full-active mesh mode and deploys QoS-aware routing to transmit packets with significantly decreased end-to-end delays as shown in Figure 11.

C. Overhead analysis

We evaluate the overhead of the mode switch scheme by measuring the duration of the deconstruction and re-construction process. In this set of experiments, we study the impact of IEEE 802.15.4 protocol parameters on the switch overhead. Primarily, the (BO,SO) has been explored at different combinations that keep the same duty-cycle as

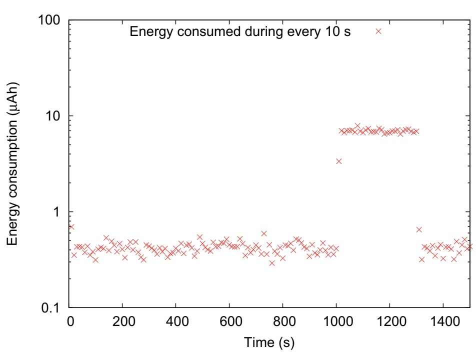


Figure 10. Energy consumption at Node 9 operating in two modes

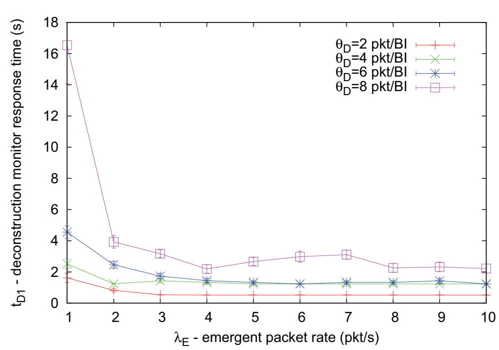
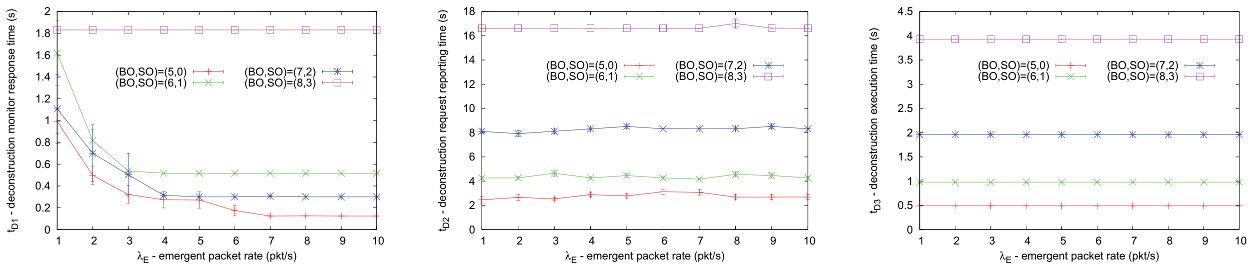


Figure 12. t_{D1} vs. deconstruction monitor threshold θ_D



(a) t_{D1} vs. λ_E (b) t_{D2} vs. λ_E (c) t_{D3} vs. λ_E

Figure 13. Deconstruction process in three steps

previously used one of (6,1). The reason is to maintain the same bandwidth distribution ratio and a similar level of the energy consumption for routine reporting in CTR.

The deconstruction process can be divided into three phases: t_{D1} is defined as the duration from the time when the emergent event starts to the time when the first coordinator detects the trigger event of deconstruction. We refer to this phase as deconstruction monitor response time. t_{D2} and t_{D3} correspond to stages 1 and 2 of the deconstruction process (cf. Section IV), respectively. t_{D2} is defined as the period from the generation time of the first DRC to the time when the PAN coordinator receives the first DRC. t_{D3} denotes the duration of the deconstruction execution process and is equal to T_{d_s2} that has been derived to be bounded by a BI (cf. Section IV). Figure 12 illustrates for a fixed (BO,SO)=(6,1) how t_{D1} is affected by the threshold θ_D for different λ_E. When $\lambda_E < 4pkt/s$, t_{D1} is strongly affected by θ_D, which determines the sensitivity of the deconstruction monitor. As λ_E increases, t_{D1} converges to a constant value that belows 2s at a wide range of θ_D between 2 and 6. Figure 13(a) further reveals the relation between t_{D1} and the (BO,SO).

The results shown in Figure 13(b) indicate that for fixed (BO,SO) t_{D2} is not sensitive to λ_E, because DRC is always transmitted first prior to the emergent packets and, thus, does not suffer long queuing delay. The measured t_{D2} validates our analytical estimation. For example, at (BO,SO)=(6,1), l_{BI} is equal to 0.983 s and l_{SD} equals to 0.03 s. We have also observed that in almost all cases, especially when λ_E is high, the first DRC will be generated either by Node 21 or Node 22, which are closest to the sources of emergent data. Thus, with $m = 5$, the measured mean of t_{D2} at around 4.4 s is smaller than the boundary of 4.756 s, which can be calculated using Eq.2 for (BO,SO)=(6,1). In addition, owing to our DRC transmission algorithm, we have observed that there are only a few DRC packets (less than 5 in the worst case) that have been generated during this phase. Compared to the high emergent traffic, such a small DRC flow has negligible impact on the concurrent data transmission.

Figure 13(c) validates the design of the DB transmission that avoids beacon collision and limits t_{D3} to a constant value of one BI.

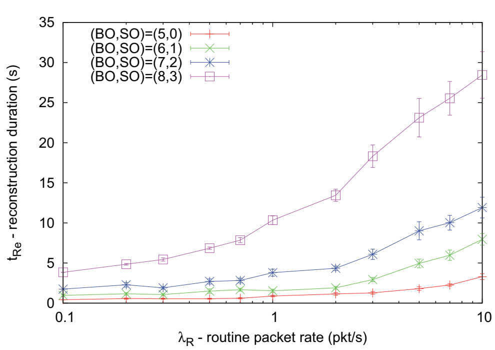


Figure 14. t_{Re} vs. λ_R under various (BO,SO)

The results in Figure 14 reveal how the routine traffic λ_R affects the reconstruction execution time t_{Re}. According to our algorithm, t_{Re} is be bounded by one BI if no beacon collision occurs. The much higher reconstruction time for large λ_R values is caused by the increased probability of beacon collisions. However, such a high traffic is unrealistic in real applications, because it exceeds the capacity of the cluster-tree network operating at a low duty-cycle. Our experiments have shown that the PDR of the routine packets can reach only $36.06\,\%$ for $\lambda_R = 1$ pkt/s and will rise to $95\,\%$ for $\lambda_R = 0.33$ pkt/s. Thus, as long as the routine traffic is low relative to the network capacity ($\lambda_R < 0.33pkt/s$), the mean of t_{Re} is very close to a BI.

VI. Conclusion

We have proposed a low-cost adaptive scheme, called "mode-switch", that supports (1) an energy-efficient mode for transmitting routine sensing data and (2) a QoS-aware low-latency mode for transmitting high-priority emergency data. It provides fast and efficient switching between the two operation modes adaptively to the sensing environment. The energy-efficiency mode is adopted for transmitting routine data in normal situations, while the low-latency mode is immediately selected for better QoS if there is emergency data. "mode-switch" supports both cluster-tree topology and mesh topology in IEEE 802.15.4 standard.

It enables on-the-fly deconstruction and reconstruction of cluster trees for mode switching. Its cross-layer design integrates network-layer and MAC-layer protocols seamlessly to meet different QoS and energy-efficiency requirements. Extensive simulations have demonstrated that "mode-switch" can achieve the desired levels of QoS and energy-efficiency with the two operation modes adaptively. It can perform autonomic switching between the two operation modes in a timely manner with minimum overheads.

REFERENCES

[1] "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low Rate Wireless Personal Area Networks (WPANs)," IEEE, IEEE Sandard 802.15.4-2006, 2006.

[2] P. Baronti, P. Pillai, V. W. Chook, S. Chessa, A. Gotta, and Y. F. Hu, "Wireless Sensor Networks: a Survey on the State of the Art and the 802.15.4 and ZigBee Standards," *Elsevier Computer Communications*, vol. 30, no. 7, pp. 1655–1695, May 2007.

[3] I. Dietrich and F. Dressler, "On the Lifetime of Wireless Sensor Networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 1, pp. 1–39, February 2009.

[4] C. Suh, Z. H. Mir, and Y.-B. Ko, "Design and implementation of enhanced IEEE 802.15.4 for supporting multimedia service in Wireless Sensor Networks," *Computer Networks*, vol. 52, no. 13, pp. 2568 – 2581, 2008.

[5] J.-S. Lee, "An Experiment on Performance Study of IEEE 802.15.4 Wireless Networks," in *10th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2005)*, vol. 2. Catania, Italy: IEEE, September 2005, pp. 451–458.

[6] J. Zheng and M. J. Lee, "A Comprehensive Performance Study of IEEE 802.15.4," in *Sensor Network Operations*. IEEE, 2006, pp. 218–237.

[7] T. H. Kim and S. Choi, "Priority-Based Delay Mitigation for Event-Monitoring IEEE 802.15.4 LR-WPANs," *IEEE Communications Letters*, vol. 10, no. 3, pp. 213–215, March 2006.

[8] A. Koubaa, M. Alves, and E. Tovar, "GTS Allocation Analysis in IEEE 802.15.4 for Real-time Wireless Sensor Networks," in *20th IEEE International Parallel and Distributed Processing Symposium (IPDPS 2006)*. IEEE, April 2006.

[9] ——, "i-GAME: An Implicit GTS Allocation Mechanism in IEEE 802.15.4 for Time-Sensitive Wireless Sensor Networks," in *18th Euromicro Conference on Real-Time Systems (ECRTS 2006)*. IEEE, July 2006, pp. 183–192.

[10] V. T. Raisinghani and S. Iyer, "Cross-layer design optimizations in wireless protocol stacks," *Computer Communications*, vol. 27, no. 8, pp. 720 – 724, 2004.

[11] S. Cho and A. Chandrakasan, "Energy efficient protocols for low duty cycle wireless microsensor networks," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, vol. 4. Salt Lake City, UT: IEEE Computer Society, May 2001, pp. 2041–2044.

[12] J.-J. Xiao, S. Cui, Z.-Q. Luo, and A. Goldsmith, "Joint estimation in sensor networks under energy constraints," in *1st IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON 2004)*, Santa Clara, CA, October 2004, pp. 264 – 271.

[13] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks," in *International Conference on Distributed Computing Systems*, Providence, Rhode Island, 2003.

[14] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He, "RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks," in *8th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 2002)*, San Jose, CA, September 2002, pp. 55–66.

[15] E. Felemban, C.-G. Lee, and E. Ekici, "MMSPEED: Multipath Multi-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, pp. 738–754, June 2005.

[16] S. C. Ergen and P. Varaiya, "Energy efficient routing with delay guarantee for sensor networks," *Wireless Networks*, vol. 13, no. 5, pp. 679–690, October 2007.

[17] E. Gelenbe and E. C. H. Ngai, "Adaptive Random Re-Routing for Differentiated QoS in Sensor Networks," in *BCS International Academic Conference*. London, UK: British Computer Society, September 2008, pp. 343–354.

[18] E. Gelenbe and E. Ngai, "Adaptive QoS Routing for Significant Events in Wireless Sensor Networks," in *5th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2008)*. Atlanta, GA: IEEE, September 2008, pp. 410–415.

[19] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, "Optimizing Sensor Networks in the Energy-Latency-Density Design Space," *IEEE Transactions on Mobile Computing*, vol. 1, pp. 70–80, January-March 2002.

[20] M. Strasser, A. F. Meier, K. Langendoen, and P. Blum, "Dwarf: Delay-aWAre Robust Forwarding for Energy-Constrained Wireless Sensor Networks," in *3rd IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS 2007)*, 2007, pp. 64–81.

[21] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher, "Real-Time power-aware routing in sensor networks," in *14th IEEE International Workshop on Quality of Service (IWQoS 2006)*. New Haven, CT: IEEE, June 2006, pp. 83–92.

[22] Y. Li, W. Ye, and J. Heidemann, "Energy and Latency Control in Low Duty Cycle MAC Protocols," in *IEEE Wireless Communications and Networking Conference (WCNC 2005)*, New Orleans, LA, March 2005, pp. 676–682.

[23] F. Chen, N. Wang, R. German, and F. Dressler, "Performance Evaluation of IEEE 802.15.4 LR-WPAN for Industrial Applications," in *5th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (IEEE/IFIP WONS 2008)*. Garmisch-Partenkirchen, Germany: IEEE, January 2008, pp. 89–96.