A MAC Protocol for Delay-Bounded Applications in Wireless Sensor Networks

Guido Di Bacco[†], Tommaso Melodia[‡], Francesca Cuomo[†]

[†] University of Rome “La Sapienza”, Rome, ITALY
[‡] Georgia Institute of Technology, Atlanta, GA

Abstract— **The problem of scheduling packet transmissions for data gathering in wireless sensor networks is studied in this paper. A scenario is considered where different sources contemporarily sense an event and signal the acquired information to a sink. Energy-latency tradeoffs for data gathering in sensor networks are explored by means of Integer Linear Programming Formulations. The objective of the optimization problems defined is to find *minimum latency* and *minimum energy* optimal data delivery trees, which are defined as aggregates of flows from multiple sources to a single receiver.**

A new distributed MAC protocol explicitly designed for Delay-Bounded Applications in Wireless Sensor Networks (DB-MAC) is also introduced. The primary objective of DB-MAC is to minimize the latency for delay bounded applications. Energy consumption is also reduced by means of a path aggregation mechanism that improves path sharing. Simulation results show that DB-MAC reduces the latency up to 70% with respect to a CSMA/CA MAC protocol, with up to 60% less transmissions. The performance of DBMAC is shown to be closer than CSMA/CA to the optimal values of latency and energy consumption.

I. INTRODUCTION

Wireless Sensor Networks (WSN) are envisioned to be developed for a wide range applications. A WSN is composed of a large number of sensor nodes, which are densely deployed either inside the phenomenon or very close to it [1]. Each sensor node is equipped with a sensing device, a low computational capacity processor, a short-range wireless transmitter-receiver and a limited battery-supplied energy. Sensor nodes monitor some surrounding environmental phenomenon, process the data obtained and forward these data towards a base station (also called *sink*). These characteristics of a WSN motivate a MAC that is different from traditional wireless MACs, since energy saving and self-configuration become the primary design constraints. Moreover, when monitoring phenomena such as fire alarms, calamities, etc., strict bounds on delay must be guaranteed. Thus, a MAC protocol for these applications needs to include mechanisms to bound the latency, and still be energy-efficient.

Ultra low power MAC protocols for WSNs have drawn the attention of researchers in the last few years. Important proposals include [2], [3], [4], [5], while several optimizations and alternatives to these protocols are also available in the literature. The standardized IEEE 802.11 Distributed Coordination Function (DCF) [6] is an example of a contention-based random access protocol, and is mainly built on MACAW [7]. It is widely considered an enabling technology for ad-hoc wireless networks because of its simplicity and robustness to the hidden and exposed terminal problems. However, recent works [8] pointed out that the energy consumption of 802.11 is not negligible when nodes are in idle mode. In [2], the authors propose a modified version of 802.11 Carrier Sensing Multiple Access (CSMA) called S-MAC, which combines Time Division Multiple Access (TDMA) scheduling with CSMA/CA contention-based medium access, without a strict requirement for time synchronization. In S-MAC nodes alternate between sleep and listen states. S-MAC is not suitable for delay bounded applications because sleep periods are in the order of milliseconds. Although this value could be tuned for different applications, the authors propose an average sleep delay of hundreds of milliseconds, which could affect the performance of delay-bounded applications. A variant of S-MAC, T-MAC, incorporates variable sleep schedules to further decrease the energy consumption [9]. Idle mode is also studied in PAMAS [3], which tries to avoid overhearing of neighboring nodes. In [5] the authors propose different configurations of CSMA/CA and propose an adaptive rate control mechanism.

TDMA based protocols have also been proposed [10], [11], where contention-free medium access is aimed at

This work was partially supported by the Italian FIRB Project VICOM.

minimum energy expenditure. However, many practical problems must be solved before TDMA can be widely used in WSNs, including synchronization and scheduling. In the LEACH protocol [4], contention-free TDMA communications are used inside dynamically formed clusters.

Data aggregation is also deemed to be an important mechanism to improve the performance of sensor networks. Recent studies [12], [13] show that data aggregation can help eliminating data redundancy and reducing the communication load. Benefits of data aggregation have been confirmed both theoretically [14] and experimentally [15].

In this paper, we study the problem of scheduling packet transmissions for data gathering in wireless sensor networks. We consider a scenario where different sources contemporarily sense an event and signal the acquired information to a sink. Energy-latency tradeoffs for data gathering are explored by means of Integer Linear Programming Formulations. The objective of the optimization problems defined is to find *minimum latency* and *minimum energy* optimal data delivery trees, which are defined as aggregates of flows from multiple sources to a single receiver. Energy Latency Tradeoffs in sensor networks have been previously studied in [16], where the authors focus on minimizing the energy dissipation given a time constraint by leveraging the modulation scaling technique, while the actual problem of finding a minimum latency data delivery tree and a minimum latency scheduling given the tree is not dealt with.

We also introduce a new distributed MAC protocol designed for Delay-Bounded Applications in sensor networks (DB-MAC). The multiple access is based on a RTS/CTS/DATA/ACK handshaking, but the protocol is improved with two mechanisms to reduce latency and save energy. In particular, DB-MAC implements an access with *priority* and *path aggregation*.

The following of the paper is organized as follows. In Section II, we discuss and formulate different optimization problems related to scheduling data transmissions in sensor networks. In Section III we introduce our proposed DB-MAC protocol and describe it in detail. In particular, in Section III-B we discuss the adopted contention scheme, while in Section III-C we describe the path aggregation mechanism in DB-MAC. In Sections IV and V latency and energy performance results are discussed, while Section VI gives the main conclusions.

II. PROBLEM FORMULATION

In this section we formulate the two optimization problems discussed in this paper. First, we introduce the *minimum latency data delivery* problem and formalize it as a sequence of Integer Linear Programs (ILP). Then, we introduce the *minimum energy data delivery* problem and again give an ILP based formulation of the problem.

A. Network Model

We represent the network of sensor nodes as an *undirected geometric random graph* $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, .., v_N\}$ is a finite set of radio nodes in a finite-dimension terrain, with $N = |\mathcal{V}|$, and $\mathcal{E}$ is the set of links among nodes, i.e., $e_{ij} = 1$ iff v_i is within v_j's transmission range and viceversa. A graph is referred to as a *random graph* when the nodes are placed randomly within a given space. A geometric random graph is a random graph where the N nodes are placed independently and uniformly and there exists a link between any two nodes if their distance is smaller than a given transmission range T_R. Moreover, D is the matrix whose element d_{ij} contains the distance between nodes v_i and v_j (i and j for simplicity in the following), while the matrix C contains elements c_{ij} that represent the cost of the link (i, j), according to the energy model that will be introduced in section II-B. Let $\mathcal{S}$ be the set of traffic sources, with $S = |\mathcal{S}|$. This set represents the sensor nodes that detect the event, i.e., the sensors that are in the event area or whose measuring match the query received from the sink [1]. Let $\mathcal{R}$ be the set of sink nodes that collect the traffic, with $N_R = |\mathcal{R}|$. For the sake of simplicity, in the following we will assume that only one sink exists, and that the sink is not a sensor node. Thus, since the set of sources is disjoint from the set of sinks we have $\mathcal{R} \subset \mathcal{V}$, $\mathcal{S} \subset \mathcal{V}$ and $\mathcal{R} \cap \mathcal{S} = \emptyset$. Although the problem formulation discussed in this section is valid for multiple sinks, studying the behavior of sensor networks with multiple sinks is out of the scope of this paper.

B. Energy Model

An accurate model for energy consumption per bit at the physical layer is $E_{elec}^{trans} + \beta d^{\alpha} + E_{elec}^{rec}$, where E_{elec}^{trans} is a *distance independent* term that takes into account overheads of transmitter electronics (PLLs, VCOs, bias currents, etc) and digital processing; E_{elec}^{rec} takes into account the overhead of receiver electronics, and βd^n accounts for the radiated power necessary to transmit over a distance d between source and destination. As in [4], we will assume that $E_{elec}^{trans} = E_{elec}^{rec} = E_{elec}$; thus the overall expression simplifies to $2 \cdot E_{elec} + \beta d^{\alpha}$, where:

$2 \leq \alpha \leq 5$ is the path loss;

β is a constant $[Joule/(bits \cdot m^{\alpha})]$;

E_{elec} is the energy needed by the transceiver circuitry to transmit or receive one bit $[Joule/bits]$.

C. Minimum Latency Data Delivery: Problem Definition and Integer Linear Programming Formulation

Our objective is to find the *Minimum Latency Data Delivery Tree* (dd-tree), i.e., a *loop-free subgraph* of $\mathcal{G}$ such that data can be relayed through a multi-hop path from the source nodes to the sink with a *minimum latency packet scheduling*. The dd-tree is described by a matrix x whose element x_{ij} equals 1 iff link (i, j) is part of the dd-tree x. A *TDMA scheduling* is defined as a set $\mathcal{T} = \{1..T_{MAX}\}$ of time slots, and a set $L = \{l^1, l^2, .., l^{T_{MAX}}\}$ of T_{MAX} *transmission matrixes*, each describing the transmissions on one time slot. Each element of the transmission matrix is associated to a link of the graph, while each matrix is associated to a time slot. Element $l^t_{ij} = 1$ if a transmission occurs on link (i, j) during time slot t or if a transmission has occurred in a previous time slot. Thus, $l^t_{ij} - l^{t-1}_{ij} = 1$ iff node i transmits to node j during time slot t. On each transmission matrix, a set of MAC constraints $\mathcal{M}$ has to be satisfied, in order for transmissions not to collide with each other.

We can introduce the following:

Problem 1: **$\mathbf{T_{MAX}}$-Data Delivery Tree Satisfiability**.

Given: $\mathcal{G}(\mathcal{V}, \mathcal{E})$, $\mathcal{S}$, r, T_{MAX}, $\mathcal{M}$

Question: Does a TDMA scheduling scheme exist which, in at most T_{MAX} time slots, transmits the data generated by the sources in $\mathcal{S}$ to the receiver r on a dd-tree built on links of $\mathcal{G}$, given the MAC constraints in $\mathcal{M}$?

Output: (Yes/No, x, L)

If a graph and a TDMA scheme can be found which answer Yes to the previous problem, the next step is finding the *energy efficient* (minimum energy) tree among those who admit a TDMA scheduling to deliver data from the sources to the sink in T_{MAX} time slots. We can thus introduce the following:

Problem 2: **Minimum Energy $\mathbf{T_{MAX}}$-Data Delivery Tree**.

Given: $\mathcal{G}(\mathcal{V}, \mathcal{E})$, $\mathcal{S}$, r, T_{MAX}, $\mathcal{M}$, C

Question: If $\mathcal{G}$ answers YES to **$\mathbf{T_{MAX}}$-Data Delivery Tree Satisfiability**, what is the minimum cost tree x on $\mathcal{G}$?

Output: (Yes/No, x, L, cost(x))

In the following, we formulate Problem 2 as an *Integer Linear Programming* (ILP) problem. The problem is formulated as sending data from a single source to a set of receivers on a multi-cast tree, which as can be easily seen is equivalent to the data delivery tree problem we are dealing with. For the sake of clarity, we will refer to the source of the multi-cast tree as the receiver and to the receivers in the multi-cast tree as the sources. Thus, in the resulting scheduling *sensor sources* receive data while the sink generates the datum to be transmitted to the sources.

The set of input parameters of the problem is:

$\mathcal{G}$: is the graph describing the network.

$\mathcal{S}$: is the set of sources.

r: is the receiver node.

c_{ij}: is the cost of the link between nodes i and j, i.e., $2 \cdot E_{elec} + \beta d^{\alpha}_{ij}$, where d_{ij} is the (i, j)-th element of the matrix D.

$\mathcal{T}$=1..T_{MAX}: is the set of time slots of the problem.

The following are the decision variables of the optimization problem:

$x_{ij} = 1$ iff link (i, j) is part of the dd-tree.

$f^k_{ij} = 1$ iif link (i, j) belongs to the flow between the receiver and the k-th source.

l^1 , l^2 ,.., $l^{T_{MAX}}$ is the set of transmission matrixes. The problem can now be formulated as:

Minimum Energy $\mathbf{T_{MAX}}$-Data Delivery Tree

Minimize:

$$C^{TOT} = \sum_{(i,j)\in\mathcal{E}} x_{ij} \cdot c_{ij} \tag{1}$$

Subject to:

$$\sum_{j\in\mathcal{V}} (f^s_{rj} - f^s_{jr}) = 1, \forall s \in \mathcal{S}, \forall r \in \mathcal{R}; \tag{2}$$

$$\sum_{j\in\mathcal{V}} (f^s_{sj} - f^s_{js}) = -1, \forall s \in \mathcal{S}; \tag{3}$$

$$\sum_{j\in\mathcal{V}} (f^s_{ij} - f^s_{ji}) = 0, \forall s \in \mathcal{S}, \forall i \in (\mathcal{V} \setminus \mathcal{R}) \ s.t. \ i \neq s; \tag{4}$$

$$f^s_{ij} \leq x_{ij}, \forall s \in \mathcal{S}, \forall i \in \mathcal{V}, \forall j \in \mathcal{V}; \tag{5}$$

$$x_{ij} \leq e_{ij}, \forall i \in \mathcal{V}, \forall j \in \mathcal{V}; \tag{6}$$

$$l^t_{ij} \leq x_{ij}, \forall t \in \mathcal{T}, \forall i \in \mathcal{V}, \forall j \in \mathcal{V}; \tag{7}$$

$$l_{ij}^{T_{MAX}} = x_{ij}, \forall i \in \mathcal{V}, \forall j \in \mathcal{V}; \tag{8}$$

$$\sum_{t=1}^{T_{MAX}} \sum_{i \in \mathcal{V}} l_{is}^{t} \geq 1, \forall s \in \mathcal{S}; \tag{9}$$

$$l_{ij}^{t-1} \leq l_{ij}^{t}, \forall i \in \mathcal{V}, \forall j \in \mathcal{V}, \forall t \in \mathcal{T} \; s.t. \; t > 1; \tag{10}$$

$$\sum_{j \in \mathcal{V}} (l_{ij}^{t} - l_{ij}^{t-1}) + \sum_{j \in \mathcal{V}} (l_{ji}^{t} - l_{ji}^{t-1}) \leq 1$$

$$\forall i \in \mathcal{V}, \forall t \in \mathcal{T} \; s.t. \; t > 1; \tag{11}$$

$$\sum_{j \in \mathcal{V}} (l_{ij}^{1} + l_{ji}^{1}) \leq 1, \forall i \in \mathcal{V}; \tag{12}$$

$$\sum_{j \in \mathcal{V}} (l_{ij}^{t} - l_{ij}^{t-1}) \leq \sum_{j \in \mathcal{V}} l_{ji}^{t-1},$$

$$\forall i \in (\mathcal{V} \setminus \mathcal{R}), \forall t \in \mathcal{T} \; s.t. \; t > 1; \tag{13}$$

$$\sum_{j \in \mathcal{V}} l_{ij}^{1} = 0, \forall i \in (\mathcal{V} \setminus \mathcal{R}); \tag{14}$$

Constraints (2)(3)(4) express conservation of flows [17]. Constraint (5) imposes that flows are transmitted on links that belong to the dd-tree, while constraint (6) imposes that flows are built on existing physical links, and constraint (7) restricts transmissions to links of the tree. Constraint (8) imposes that in the last time slot, data has been transmitted on all the links composing the tree, while constraint (9) imposes that each source receives the data. Constraint (10) imposes that the links included in the transmission matrix during time slot $t-1$ are a subset of those in time slot t, according to the definition given in this section. Eq. (11) imposes the MAC constraints $\mathcal{M}$, i.e., each sensor can only transmit to or receive from one node in each time slot. Eq. (12) is the equivalent of eq. (11) for the first time slot. Finally, constraint (13) imposes that in order for a node j to transmit in time slot t, the node must have received the datum in the previous $t-1$ time slots, while constraint (14) states that only the receiver can transmit in the first time slot.

The next step in solving our problem is finding the tree that can deliver data from the sources to the sink in the minimum number of time slots, thus finding the *minimum latency* (minimum number of time slots needed), the *minimum latency tree* and the *minimum latency scheduling*. We will refer to this problem as **OPTL** in the following.

Problem 3: **Minimum Latency Data Delivery Tree (OPTL)**

Given: $\mathcal{G}(\mathcal{V}, \mathcal{E})$, $\mathcal{S}$, r, $\mathcal{M}$, C

Question: What are the minimum latency T, the minimum latency scheduling L, the minimum energy given minimum latency tree x, the minimum energy cost given minimum latency?

Output: (T*,*x*,*L*,cost)*

The solution to Problem 3 can be obtained by solving Problem 2 with increasing number of time slot. The first satisfiable instance defines the minimum latency, and the output of 2 for the first satisfiable instance defines the minimum scheduling and the minimum energy tree given minimum scheduling, i.e., we find the energy efficient tree among those with minimum latency.

It is intuitive that a lower bound on the latency is given by the minimum number of hops that separate the farthest source from the sink. We refer to this number as max_min_path. It is also intuitive that after t time slots, at most

$$Y = \sum_{i=1}^{t-1} 2^{i} \tag{15}$$

sources can be reached from the sink. Thus, for the scheduling to be feasible in t time slots, we require that the number of sources S be lower than Y. Thus, we can set a lower bound for T as $T \geq max(Y, max\_min\_path)$.

Algorithm 1 can be applied to solve the given problem. At the end of its execution, T is the minimum latency, x is the minimum energy tree given minimum latency, L is the minimum latency TDMA scheduling and $cost$ is the energy cost of the dd-tree.

It can be shown that an ILP problem is at least as complex as the Geometric Connected Dominating Set problem, which is proven to be NP-complete [18]. Hence, **Minimum Energy $\mathbf{T_{MAX}}$-Data Delivery Tree** is NP-complete. However, it is still possible to solve the optimization problem for networks of moderate size with the following goals:

1) gaining some insight on the properties of the optimal solution;
2) use the optimal solution as a benchmark for the performance of suboptimal, but more scalable heuristics;
3) compare the optimal solution with feasible distributed algorithms.

Algorithm 1
FIND THE MINIMUM LATENCY

```
begin
$max\_min\_path = 0$
for each source $s$ in $(S)$ do
   find number of hops $p$ in minimum hop path to $r$
   if ($p \geq max\_min\_path$) then
      $max\_min\_path = p$
   end if
end for
calculate $Y = \sum_{i=1}^{t-1} 2^i$
$t$=$max$($Y$,$max\_min\_path$)
(Answer,$x$,$L$,$cost$)=Minimum Energy $T_{max}$ Data Delivery($t$)
while Answer=NO do
   $t = t + 1$
   (Answer,$x$,$L$,cost)=Minimum Energy $T_{max}$ Data Delivery(t)
end while
$T = t$
end
```

We implemented the model of the ILP problem in AMPL [19] and solved it with CPLEX [20], which uses a branch and bound algorithm to solve mixed integer problems. We will compare the optimal results with our proposed distributed protocol in Section V-B.

D. *Minimum Latency Scheduling on Minimum Latency Tree*

To further study energy latency tradeoffs in sensor networks, we also find the minimum latency given minimum energy data delivery tree (**OPTE** in the following):

Problem 4: **Minimum Latency given Minimum Energy Data Delivery Tree (OPTE)**.
Given: $\mathcal{G}(\mathcal{V}, \mathcal{E})$, $\mathcal{S}$, r, $\mathcal{M}$, x_{min}
Question: What are the minimum latency T and the minimum latency scheduling L, given the minimum energy tree x_{min}?
Output: (T,L)

Finding a minimum energy tree (also known as Steiner Tree) is a well known problem [17] for which several ILP formulations exist. Thus, Problem 4 can be solved by finding the minimum energy tree, and by modifying the ILP formulation given in section II-C such a way that the minimum energy tree x is an input parameter of the problem and not a decision variable. Then, Algorithm 1 can be used to find minimum latency and minimum scheduling on the given tree.

III. THE DISTRIBUTED DB-MAC PROTOCOL

This section presents a distributed MAC protocol explicitly designed for Delay-Bounded Applications in WSNs (DB-MAC).
The objectives of the proposed protocol are:

- reduce the latency, e.g., the number of time slots needed to transmit MAC frames (also packets in the following) from a set of sources to a sink;
- increase the network efficiency;
- reduce the network energy consumption.

DB-MAC adopts a CSMA/CA contention scheme based on a four way RTS/CTS/DATA/ACK handshaking, and energy consumption is reduced with data aggregation, which improves path sharing.

A. *Path aggregation*

We consider a scenario where different sources contemporarily sense an event. These sources are scattered in a wide geographic area and must convoy the sensed information to a sink, trying to minimize the elapsed time. A set of flows are generated from the sources in S to the sink. Flows can be dynamically aggregated in the path towards the sink giving rise to an aggregation tree. Intermediate nodes in the path may aggregate several flows into a single flow to reduce transmissions and the amount of data to be transmitted. Data size reduction mainly depends on the nature of the data itself and on the overlying applications. Thus, data aggregation can reduce the total size of the data or not. We refer to a procedure which aggregates two packets received from different nodes, and creates a unique packet with the same length, as *aggregation with Size Reduction*. This form of aggregation can be used, for example, in sensor networks intended for temperature monitoring. When the network operator is only interested in averaged regional values of the temperature, an intermediate node that receives two values of temperature can calculate the average and forward it to the sink. The resulting data field has the same length as the incoming packets.

Aggregation without Size Reduction occurs when two packets received from different nodes are merged in a single packet with a longer data field. If we assume that a single packet is transmitted in a single time slot, in case of *aggregation with Size Reduction* the outgoing packet is transmitted in a single time slot, while *aggregation without Size Reduction* implies transmission in multiple time slots. In both cases, the MAC layer leverages data aggregation since the overall transmission overhead (e.g., MAC headers, MAC control frames) can be reduced. Furthermore, since medium contention is performed at the MAC layer for each packet to be transmitted, the contention overhead is reduced when a node contends only once to transmit a longer packet with respect to multiple contentions for shorter packets.

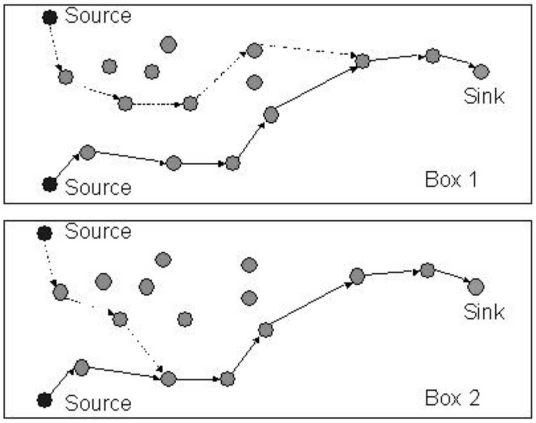


Fig. 1
AGGREGATION MODES: CLOSE TO THE SINK (BOX 1), CLOSE TO THE SOURCES (BOX 2)

Flows can be aggregated in a node that is close to the sink (Figure 1, Box 1) or in a node that is close to the sources (Figure 1, Box 2). An aggregation mechanism that explicitly tries to aggregate two (or more) flows as close as possible to the sources is more energy efficient. The total number of transmissions is reduced and, as a consequence, the latency is also decreased. In the example of Figure 1, in case of Box 1 latency and energy consumption depend on 12 transmissions, while in case of Box 2 they depend on 10 transmissions.

B. DB-MAC aggregation by exploiting RTS/CTS and priorities

The contention mechanism of DB-MAC is based on the CSMA/CA scheme. The contention procedure begins when a node senses the channel to determine whether or not another node is transmitting. The collision avoidance mechanism is based on two techniques: *InterFrame Space insertion* (IFS) and a *backoff* algorithm. The IFS is the time a node must wait before transmitting after it senses an idle channel.

If the channel is idle for a period of time equal to the DCF IFS (DIFS), the node picks a random number (named Contention Window- CW) between CW_{MIN} and CW_{MAX}, and waits for a Backoff Interval (BI) proportional to CW. A collision occurs if two or more nodes select the same BI.

With respect to the basic CSMA/CA mechanism, DB-MAC introduces two novel mechanisms:

- RTS/CTS messages are exploited to perform data aggregation;
- BIs are computed by taking into account the *priority* assigned to different transmissions.

Both these mechanisms are explicitly designed to improve the energy and latency performance of a WSN.

DB-MAC achieves an efficient path aggregation by modifying the basic RTS/CTS mechanism. Each node can take advantage of transmissions from other nodes, by overhearing CTSs, in order to facilitate an *early* data aggregation (i.e., a data aggregation close to the source). This is also achieved by exploiting transmission priorities that DB-MAC assigns to different nodes: a transmission close to the source has a higher priority than a transmission close to the sink.

By combining these two mechanisms (CTS overhearing and priorities) a node gains access to the medium with a higher probability if it is close to the source and it performs the path aggregation as close as possible to the transmitting sources.

C. The path aggregation mechanism in DB-MAC

When a node gains access to the medium and sends out the RTS packet, the receiver replies with a CTS. In DB-MAC, unlike in 802.11, the RTS and CTS packets contain the priority level in a header field. We will illustrate the path aggregation mechanism by referring to Figure 2. Two nodes A and C want to transmit packets respectively to B and D, that are relay nodes in the path towards the sink. We assume that A and C have data packets with priority 13 and 12, respectively. Since $Pr(A) < Pr(B)$, the transmitter A wins access to the medium (details on how this is achieved are in

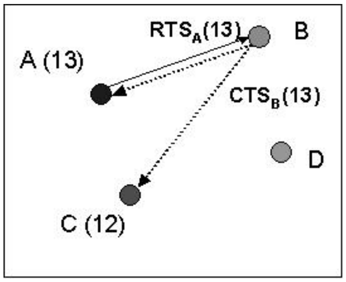


Fig. 2
THE PATH AGGREGATION MECHANISM IN DB-MAC

section III-D) and sends an RTS (Pr=13) packet to B. B replies with a CTS (Pr=13) packet. If C is within transmission range of B, it overhears the CTS (Pr=13) sent by B. Subsequently, A sends a data packet and B confirms its correct reception with an ACK.

In the following contention period, B decreases its priority to 12, while C keeps Pr=12. Nodes B and C will now contend for the channel. Suppose that node C wins the access to the medium. Since C knows that B has received a data packet which has not been forwarded yet (as C has not heard an RTS from B), C decides to use B instead of D as relay node. This produces the path aggregation at B of flows coming from A and C. After receiving the packets from A and C, B aggregates data with or without Size Reduction as explained in Section III. In both cases, a single data packet is produced.

The priority access, supported by the data aggregation mechanism, enables consistent energy saving with respect to a basic 802.11 scheme. In particular, it allows scheduling alternate transmissions of packets from different sources in scenarios with localized sources. This increments the probability of path aggregation.

D. Assigning priorities in DB-MAC

As introduced in section III-C, when a source starts transmitting the priority is set to the maximum ($Pr = Pr_{MAX}$). This value (Pr) is then decreased by one at each hop. The receiving node decrements by 1 the priority from Pr_{MAX} to $Pr_{MAX}-1$, and forwards the packet to the next node, contending for medium access with a priority $Pr_{MAX}-1$.

It is to be noticed that our priority policy differs in many respects from the IEEE 802.11e draft standard. In fact, the 802.11e priority mechanism is not suitable for WSN applications for several reasons. First, only four different priority levels are defined. Moreover, if we consider two nodes contending for medium access, with priority x and y respectively, with $x > y$, the node with priority x is not assured to win medium access. In our priority access scheme, defined in detail in the next Section, more than 4 priority levels are needed, and we want to guarantee that nodes with a higher priority always gain access to the medium before nodes with lower priorities.

Unlike in 802.11, where the BI value is set between 0 and 1023 tics, we divide the CW interval in Pr_{MAX} equal subintervals, named $CWInt(x)$, that depend on the value of the priority x (Figure 3). The tic is defined as the time unit of the MAC protocol and we set one tic equal to the *Slot Time* defined in 802.11 ($20\mu s$).

If we define:

$$CWInt(x) = [CW_{MIN}(Pr = x), CW_{MAX}(Pr = x)] \tag{16}$$

and

$$\begin{aligned} &RCW(CWInt(x)) = \\ &\quad Random[CW_{MIN}(Pr = x), CW_{MAX}(Pr = x)] \\ &\qquad \forall x \in \{1, 2, .., Pr_{MAX}\} \end{aligned} \tag{17}$$

then the subintervals can be set such that:

$$\begin{aligned} &RCW(CWInt(x)) < RCW(CWInt(y)) \\ &\qquad \forall x, y \in \{1, 2, .., Pr_{MAX}\} : x < y \end{aligned} \tag{18}$$

The length of BI, given a priority x, is calculated as follows:

$$BI(Pr = x) = RCW(CWInt(x)) * tic \tag{19}$$

As a consequence of eq. (17):

$$\begin{aligned} &BI(Pr = x) < BI(Pr = y) \\ &\qquad \forall x, y \in \{1, 2, .., Pr_{MAX}\} : x < y \end{aligned} \tag{20}$$

The backoff timer is frozen when a transmission is detected and reactivated when the channel becomes idle again. The receiving node waits for a Short IFS (SIFS) and responds with an acknowledgment (ACK) to confirm a successful transmission. The SIFS is smaller than the DIFS to allow ACKs to be transmitted immediately without entering the backoff process. The DIFS is assumed to be the same as in 802.11 (equal to $60\mu s$=3 $tics$) and the SIFS is set to $20\mu s$ ($1 tic$). Figure 3 depicts the access scheme timing.

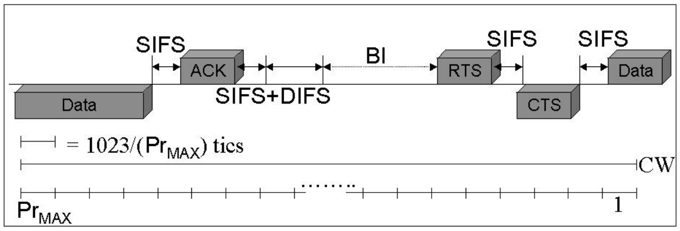


Fig. 3

THE CONTENTION MECHANISM IN DB-MAC

IV. Simulation Environment

We implemented the DB-MAC protocol in a C++ simulator which models the CSMA/CA MAC in detail. We consider the general case where a set of nodes is scattered in an area such that not all nodes are within transmission range of each other. A subset of the nodes senses the event and transmits the monitoring data. Time is slotted and a slot time is equal to the time required to transmit one data packet.

Each source generates a data packet and sets the packet priority to Pr_{MAX}. All sources start contending with the same priority (Pr_{MAX}). When two sources are within transmission range of each other only one of them gains access to the medium. The source which loses the contention waits for the channel to become idle and then participates to a new contention. Nodes that receive a packet from a source at the end of the first slot decrement the packet priority by 1 (from Pr_{MAX} to $Pr_{MAX} - 1$) and contend for the second slot. Then, the node with the highest priority (among all transmitters) gain the medium. For the sake of simplicity, possible collisions among RTS/CTS packets are not considered in our simulation model. In fact, if the network is not congestioned, the contribution of RTS/CTS collisions to the overall latency and energy consumption is not significant.

Since our objective is to evaluate the performance at the MAC level, we do not implement any particular routing algorithm proposed for WSNs. In order to determine the path from each source to the sink, we calculate the optimal path for each flow with the Bellman-Ford algorithm. The link metric used is the same as the energy model described in section II.

We compare three different protocols:

1) basic CSMA/CA;
2) CSMA/CA with Path Aggregation (PA);
3) DB-MAC.

CSMA/CA is considered as a reference protocol. We assume that if a backlogged node receives a second data packet to be forwarded, it merges the content of the two packets and forwards only one packet (*occasional path aggregation*). We also simulated the CSMA/CA mechanism improved with an *intentional path aggregation* mechanism (named CSMA/CA with PA), as discussed in Section III. Finally, we test the full DB-MAC protocol, which uses the intentional path aggregation mechanism and contends the medium with the priority mechanism explained in Section III.

V. Results and Discussions

In this section we discuss performance results of DB-MAC and compare them with the performance of CSMA/CA and CSMA/CA with PA. Then, we compare DB-MAC to the solution of the optimization problems discussed in Section II.

A. Performance evaluation of DB-MAC

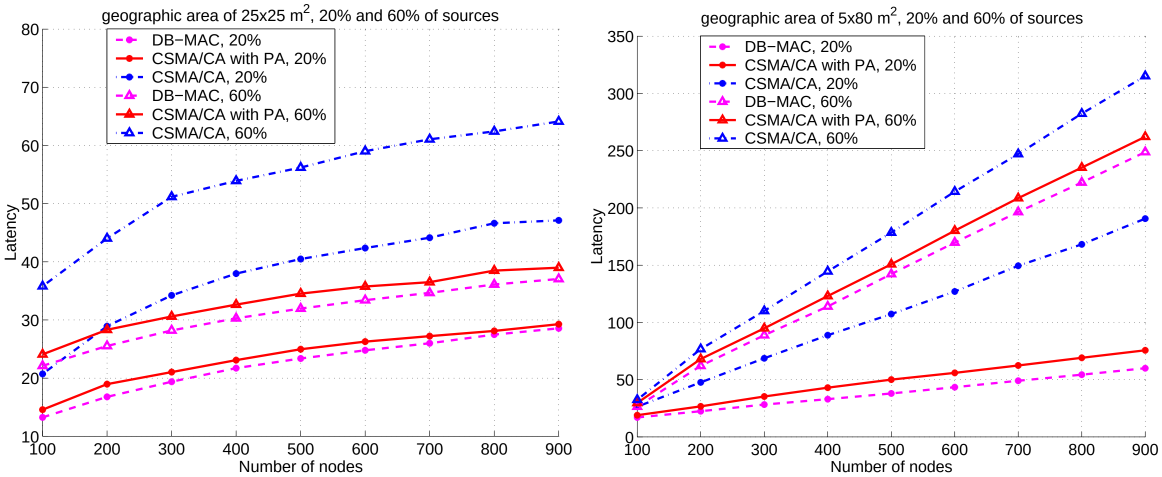
The choice of the Pr_{MAX} parameter in Section III-B needs further discussion. If Pr_{MAX} is too small with respect to the maximum number of hops in a path, from the $(Pr_{MAX}+1)-th$ hop to the sink it is not possible to exploit the priority mechanism discussed in Section III-B. On the other hand, the higher the value of Pr_{MAX}, the smaller is $CWInt(x)$. As $CWInt(x)$ decreases, the probability that a collision occurs increases. In all the simulations performed we found paths with a maximum number of hops comprised between 4 and 7. With the aim of overprovisioning the value of Pr_{MAX}, we set Pr_{MAX}=16.

The simulation results presented deal with two different scenarios. In the first, sensor nodes are uniformly distributed in a geographic area of 25×25 m^2. The sources and the sink are randomly selected among the sensor nodes. Since sources are scattered in the entire terrain, the monitored phenomenon is not concentrated in a limited area. This represents the monitoring of a distributed event. A transmission range of 8 meters is considered.

The second scenario refers to a localized event to be monitored far from the sink. The geographic area considered is 5×80 m^2 and the sources are uniformly distributed in a square area of 5×5 m^2 on one side, while the sink is at the opposite side. Such scenario allows evaluating the performance of the MAC protocols in case of long multi-hop paths. The number of sources is set to 20% and 60% of the total number of nodes, respectively.

For both scenarios, we evaluate the latency (Figure 4) as a function of the number of nodes for CSMA/CA, CSMA/CA with PA and DB-MAC. The latency is defined as the number of time slots to transmit a data packet from all the sources to the destination.

We define the *Decrement of Latency* as the difference between the latency of DB-MAC (or CSMA/CA with Path Aggregation) and CSMA/CA, normalized to the latency of CSMA/CA. In Figure 5 we show the Decrement of Latency for CSMA/CA with PA and DB-MAC with respect to CSMA/CA as a function of the number of nodes. This figure shows the benefits of the

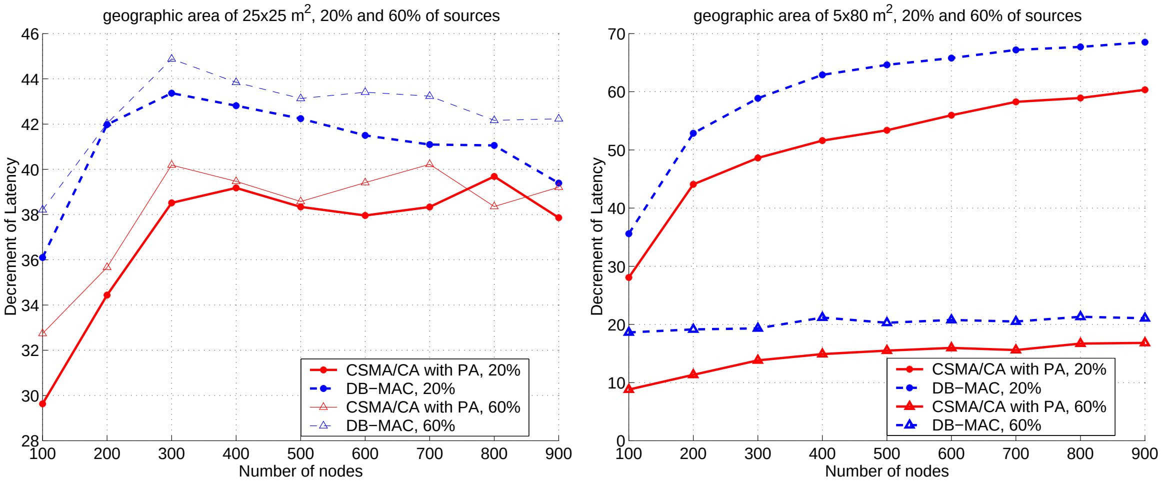


(a) Latency in a 25×25 m^2 scenario

(b) Latency in a 5×80 m^2 scenario

Fig. 4

LATENCY EVALUATION IN SCENARIOS WITH DISTRIBUTED AND LOCALIZED SOURCES



(a) Decrement of Latency in a 25×25 m^2 scenario

(b) Decrement of Latency in a 5×80 m^2 scenario

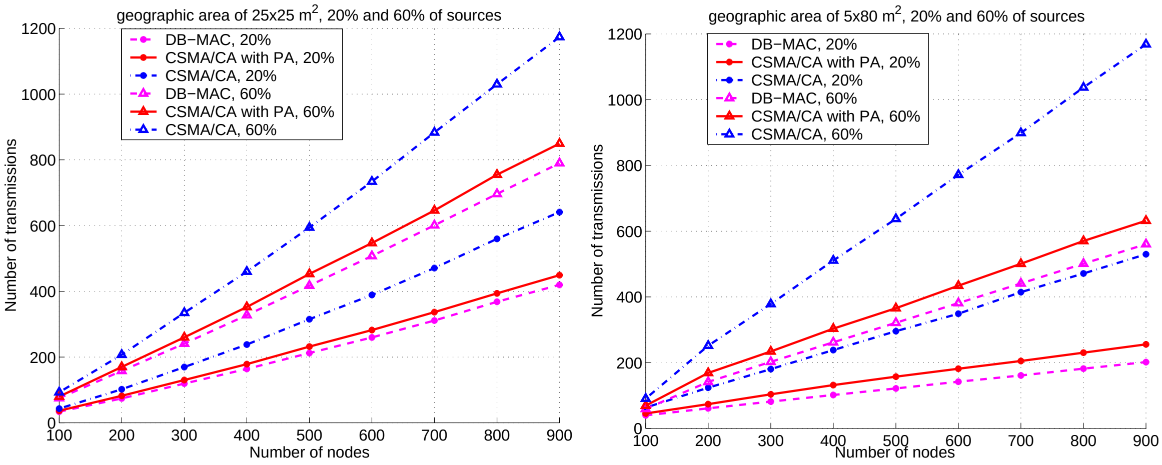
Fig. 5

DECREMENT OF LATENCY IN SCENARIOS WITH DISTRIBUTED AND LOCALIZED SOURCES

path aggregation mechanism.

indent The main differences between scenario 1 and 2 is the average length of data paths. As a consequence, the latency in Figure 5(b) is higher than in Figure 5(a). However, in both cases DB-MAC reduces the latency with respect to CSMA/CA and CSMA/CA with Path Aggregation. Since CSMA/CA with PA shows decreased latency with respect to CSMA/CA, we can conclude that both mechanisms (Path Aggregation and priority contention) improve latency.

Referring to Figure 4(b), when sources are localized, in case of contention with priority, the Decrement of Latency (up to 10%) is higher than the scenario of Figure 4(a) (up to 5%). Moreover, we observe that the latency

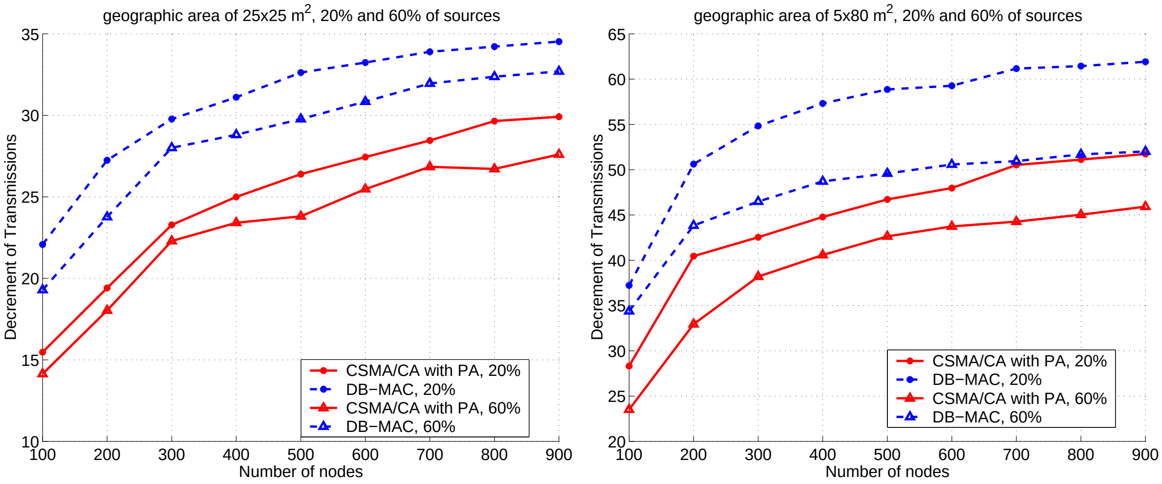


(a) Number of transmissions in a 25×25 m^2 scenario

(b) Number of transmissions in a 5×80 m^2 scenario

Fig. 6

NUMBER OF TRANSMISSIONS IN SCENARIOS WITH DISTRIBUTED AND LOCALIZED SOURCES



(a) Decrement of number of transmissions in a 25×25 m^2 scenario

(b) Decrement of number of transmissions in a 5×80 m^2 scenario

Fig. 7

DECREMENT OF NUMBER OF TRANSMISSIONS IN SCENARIOS WITH DISTRIBUTED AND LOCALIZED SOURCES

depends on the number of sources in the sensor network when the average path length increases. While in Figure 4(a) the increment in latency when the number of sources increases from 20% to 60% of the nodes is quite limited, in Figure 4(b) the difference is up to 50%.

Figure 6 shows the number of transmissions vs. number of nodes, for CSMA/CA, CSMA/CA with PA and DB-MAC. We define the *Decrement of transmissions* as the difference between the number of transmissions of DB-MAC (or CSMA/CA with Path Aggregation) and the number of transmissions for CSMA/CA, normalized to the number of transmissions for CSMA/CA. In Figure 7 we plot the Decrement of transmissions vs. number of nodes. With reference to Figures 6 and 7, we observe that both the path aggregation (CSMA/CA with Path Aggregation) and the priority access mechanism (DB-MAC) improve CSMA/CA. Moreover, the number of transmissions decreases more in the scenario with localized sources. While the number of transmissions with CSMA/CA is quite independent of the scenario, we observe a better performance with DB-MAC in the 5×80 m^2 scenario. This is confirmed by Figure 7, where the decrement of transmissions is up to 60% for 900 nodes.

B. Comparison with OPTE and OPTL

In this section we compare the performance of DB-MAC with the solution of the optimization problems presented in Section II-C, namely the *minimum latency data delivery tree* (OPTL) and the *minimum energy data delivery tree* (OPTE).

Figure 8 shows Energy-Latency tradeoffs for the considered protocols. In the simulations performed, the number of sensor nodes deployed ranges from 40 to 140, while the terrain is square sized with side variable between 10 and $50\,m$; the number of sources that detect the event is set to 10. On the x axis we report different values of normalized latency while on the y axis we report energy expenditure (in Joules). In Figures 8(a) and 8(b), simulation results for different experiments are shown for CSMA/CA and CSMA/CA with path aggregation, respectively. In average, it can be seen that the path aggregation mechanism brings both latency and energy benefits, as redundant transmissions are avoided. This saves energy and at the same time avoids useless contentions which deteriorate the latency performance. In Figure 8(c) we plot simulations results for DB-MAC, while in Figure 8(d) we report values obtained by solving OPTE and OPTL in the same scenarios. While the performance of CSMA/CA and CSMA/CA with PA is far from the optimal results, DB-MAC is close enough to OPTE and OPTL in most of the simulations. In particular, DB-MAC is close to OPTE both in terms of energy and latency, while it is slightly more distant from OPTL in terms of latency. In general, it can be seen that a OPTL solution also shows near-optimal energy consumption in some simulations, while in other optimal latency cannot be obtained without increasing the energy consumption.

Finally, in Figure 9 we compare the behavior of DB-MAC, OPTE and OPTL with 3D plots, with variable number of deployed sensor nodes and dimensions of the terrain. The number of sources is again fixed to 10. The latency value for OPTL is shown (Figure 9(a)) to be fairly independent of the number of nodes and of the size of the area. Surprisingly, higher latencies are obtained with a smaller number of nodes. OPTE shows higher values of latency, again with a fairly flat behavior. DB-MAC shows a slightly higher dependence on the number of nodes, but is still not dependent on the dimensions of the area. Figure 9(b) reports energy expenditure. As expected, OPTE, OPTL, and DB-MAC are shown to consume more energy when the dimensions of the terrain increase and not to be highly dependent on the number of nodes in the terrain.

VI. CONCLUSION AND FUTURE WORKS

In this paper, we discussed minimum latency and minimum energy data delivery in sensor networks. We defined Integer Linear Programs to find *minimum latency* and *minimum energy* optimal data delivery trees, which are defined as aggregates of flows from multiple sources to a single receiver.

We also introduced a new distributed MAC protocol explicitly designed for Delay-Bounded Applications in wireless sensor networks (DB-MAC). The protocol is based on RTS/CTS/DATA/ACK handshaking, but is improved by means of two mechanisms to reduce latency and save energy. In particular DB-MAC implements an access with priority and path aggregation. Simulations show that in the considered scenarios DB-MAC leads to a reduction up to 70% in latency and up to 60% in the number of transmissions with respect to CSMA/CA. The performance of DB-MAC is shown to be much closer than CSMA/CA to the optimal latency and energy performance.

REFERENCES

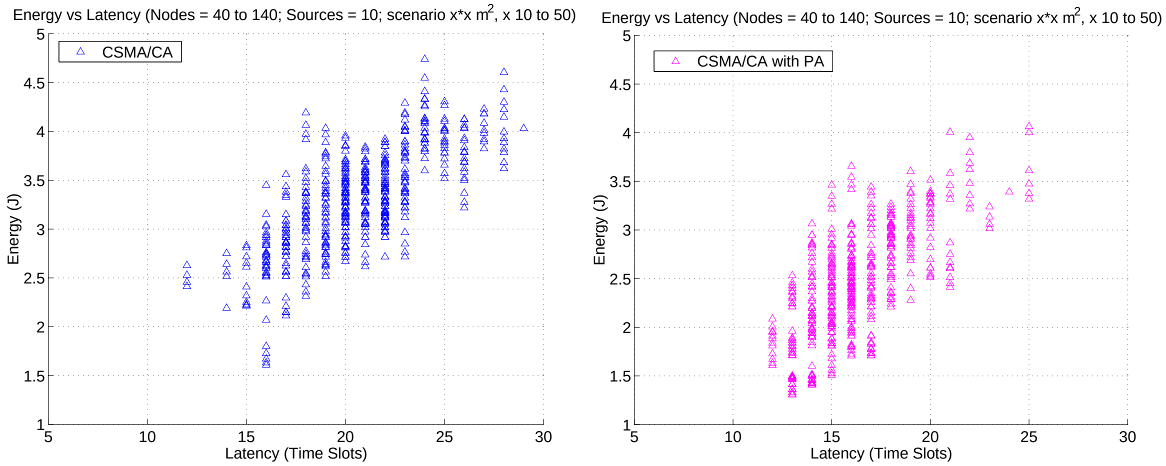
[1] I. Akyildiz,W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless Sensor Networks: A Survey, Computer Networks (Elsevier) Journal, pp. 393- 422, March 2002.

[2] W. Ye, J. Heidemann, and D. Estrin. An energy efficient mac protocol for wireless sensor networks. In Proc. of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (Infocom 2002), pages 15671576. New York, June 2002.

[3] S. Singh and C.S. Raghavendra, PAMAS: Power aware multi-access protocol with signalling for ad hoc networks, ACM Computer Communication Review, vol. 28, no. 3, pp. 526, July 1998.

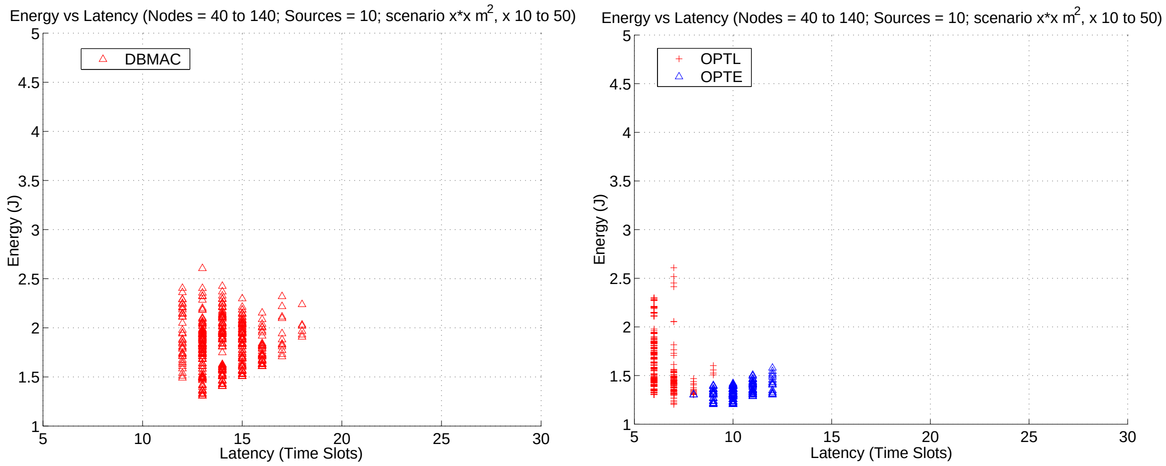
[4] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, Energy-efficient communication protocols for wireless microsensor networks, Hawaii International Conference on Systems Sciences, Jan. 2000.

[5] A. Woo and D. Culler, A transmission control scheme for media access in sensor networks, ACM/IEEE International Conference on Mobile Computing and Networking, Rome, Italy, July 2001.



(a) Latency vs Energy for CSMA/CA

(b) Latency vs Energy for CSMA/CA with PA



(c) Latency vs Energy for DB-MAC

(d) Latency vs Energy for Optimal Energy and Optimal Latency

Fig. 8

ENERGY-LATENCY TRADE-OFFS

[6] LAN MAN Standards Committee of the IEEE Computer Society, Wireless LAN medium access control (MAC) and physical layer (PHY) specification, IEEE, New York, NY, USA, IEEE Std 802.11-1997 edition, 1997.

[7] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, Macaw: A media access protocol for wireless lans, ACM SIGCOMM Conference, 1994.

[8] M. Stemm and R. H. Katz, Measuring and reducing energy consumption of network interfaces in hand-held devices, IEICE Transactions on Communications, vol. E80-B, no. 8, pp. 11251131, Aug. 1997.

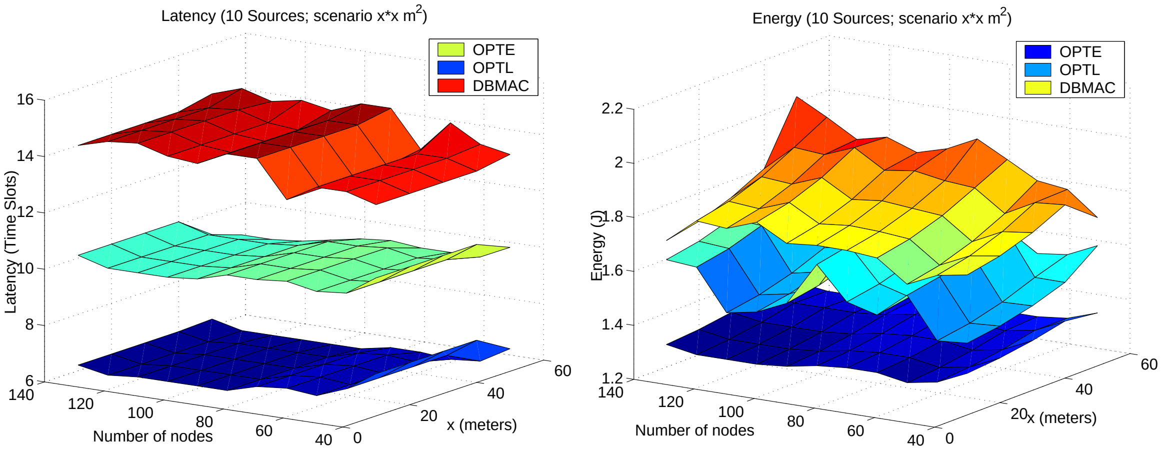
[9] T. van Dam, K. Langendoen", An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks, Proceedings of ACM SENSYS 2003, Los Angeles, CA, USA, November 2003.

[10] K. Arisha, M. Youssef, M. Younis, Energy-Aware TDMA-Based MAC for Sensor Networks, Proceedings of the IEEE Integrated Management of Power Aware Communications, Computing and Networking (IMPACCT 2002) New York City, New York, May 2002.

[11] V. Rajendran, K. Obraczka, J. J. Garcia-Luna-Aceves, Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks, Proceedings of ACM SENSYS 2003, Los Angeles, CA, USA, November 2003.

[12] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom2000), Boston, Massachusetts, August 2000.

[13] C. Intanagonwiwat, D. Estrin, R. Govindan, J. Heidemann,



(a) Latency vs dimension of the terrain and number of nodes

(b) Energy vs dimension of the terrain and number of nodes

Fig. 9

ENERGY AND LATENCY VS DIMENSION OF THE TERRAIN AND NUMBER OF SOURCES

Impact of Network Density on Data Aggregation in Wireless Sensor Networks, International Conference on Distributed Computing Systems (ICDCS-22). November 2001.

[14] J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan. Building efficient wireless sensor networks with low-level naming. ACM Symposium on Operating Systems Principles, Banff, Canada, October 2001.

[15] B. Krishnamachari, D. Estrin, and S. Wicker. Modelling data-centric routing in wireless sensor networks, IEEE Infocom 2002.

[16] Y. Yu, B. Krishnamachari, V. K. Prasanna, Energy-Latency Tradeoffs for Data Gathering in Wireless Sensor Networks, Proceedings of IEEE INFOCOM 2004, Hong Kong S.A.R., P.R. China, March 2004.

[17] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, February 1993.

[18] M. R. Garey and D. S. Johnson, Computer and Intractability, W. H. Freeman and Co., 1979.

[19] R. Fourer, D. M. Gay, B. W. Kernighan AMPL: A Modeling Language for Mathematical Programming, Duxbury Press / Brooks/Cole Publishing Company, 2002.

[20] www.cplex.com