WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
*Wirel. Commun. Mob. Comput.* 2011; **11**:1603–1617
Published online 17 November 2011 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/wcm.1222

SPECIAL ISSUE PAPER

# Localized delay-bounded and energy-efficient data aggregation in wireless sensor and actor networks

Xu Li[1], Shuo Yan[2], Chendong Xu[2], Amiya Nayak[2] and Ivan Stojmenovic[2]*

[1] INRIA Lille—Nord Europe, Villeneuve d'Ascq, France
[2] University of Ottawa, Ottawa, Ontario, Canada

## ABSTRACT

In data aggregation, sensor measurements from the whole sensory field or a sub-field are collected as a single report at an actor by using aggregate functions such as sum, average, maximum, minimum, count, deviation, and so on. We propose a localized delay-bounded and energy-efficient data aggregation (DEDA) protocol for request-driven wireless sensor networks with IEEE 802.11 carrier sense multiple access with collision avoidance run at media access control layer. This protocol uses a novel two-stage delay model, which measures end-to-end delay by using either hop count or degree sum along a routing path depending on traffic intensity. It models the network as a unit disk graph (UDG) and constructs a localized minimal spanning tree (LMST) sub-graph. Using only edges from LMST, it builds a shortest-path (thus energy-efficient) tree rooted at the actor for data aggregation. The tree is used without modification if it generates acceptable delay, compared with a given delay bound. Otherwise, it is adjusted by replacing LMST sub-paths with UDG edges. The adjustment is done locally on the fly, according to the desired progress value computed at each node. We further propose to integrate DEDA with a localized sensor activity scheduling algorithm and a localized connected dominating set algorithm, yielding two DEDA variants, to improve its energy efficiency and delay reliability. Through an extensive set of simulation, we evaluate the performance of DEDA with various network parameters. Our simulation results indicate that DEDA far outperforms the only existing competing protocol. Copyright © 2011 John Wiley & Sons, Ltd.

**KEYWORDS**
data aggregation; delay bound; wireless sensor networks

***Correspondence**
Ivan Stojmenovic, School of Electrical Engineering and Computer Science, University of Ottawa, 800 King Edward, Ottawa, Ontario K1N 6N5, Canada.
E-mail: ivan@site.uottawa.ca

## 1. INTRODUCTION

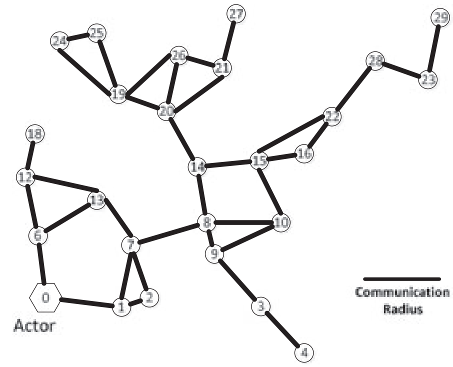
Sensors are multifunctional devices that are able to respond to physical stimuli and communicate via wireless links [1,2]. They are small in size, low in cost, and equipped with limited resources such as energy, memory, and CPU cycle. A massive number of sensors are usually dropped in a region of interest (ROI). Once deployed, they self-organize into a multi-hop ad hoc network, that is, wireless sensor network (WSN), and operate autonomously: sampling the environment and reporting the samples to one or a few data gathering actors (a.k.a. sinks) that normally have no resource limitation. In most applications, they report to the closest actor. When actors are out of range, sensors will rely on each other for forwarding data packets to the actors. When data correlation exists, data from different sources (sensors) are aggregated and combined into a single report, by using aggregate functions such as sum, average, maximum, minimum, count, deviation, and so on, at intermediate nodes to save bandwidth and energy [3].

The data aggregation paths from sensors to an actor form a reverse broadcast tree, called *data aggregation tree*, rooted at the actor. Central to the data aggregation problem is the construction of this tree. The selection of a data aggregation protocol for a given WSN is subject to the information available at the network nodes. If each node is provided with full topology information of the network and if the information is stable over time, a centralized solution can be applied. It may lead to an optimal data aggregation structure. However, if actor or sensor nodes do not know the full network graph and that network dynamics are present, centralized solutions will not work as expected because obtaining such knowledge is costly. In contrast, localized solutions only require the information available from neighbors for making protocol decisions. Although this type of protocols yield sub-optimal results, they are efficient and scalable for large-scale networks and are remarkable in practice.

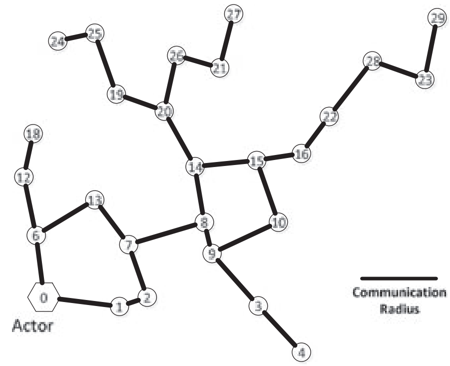
## 1.1. Motivation

Sensors are normally powered by low-energy batteries. Manual replacement or recharge of sensor batteries is infeasible most of time because of operational factors such as human inaccessibility to the sensory field or tight maintenance budget. It is therefore highly desirable to prolong the lifetime of a WSN as a whole by minimizing and balancing energy usage among individual sensors. In critical real-time scenarios such as disaster management, emergency rescue, battle field surveillance, and so on, sensor reports are often required to arrive at actors with bounded delay so as to ensure timely event response. Failure to do so may cause loss of lives and damage to economy.

Existing data aggregation protocols usually require centralized control and/or emphasize on energy efficiency. They seldom consider the delay problem. The only known localized delay-bounded power-aware data aggregation protocol [4], referred to as MS here, has major drawbacks in energy saving as well as in delay modeling, and it thus has limited effect on prolonging network lifetime and meeting delay requirement. Motivated by the insufficiency and incompleteness of previous work, in this article we address how to achieve power optimality in data aggregation while respecting a given delay constraint.




(a) Original network topology




(b) Local Minimum Spanning Tree (LMST)

**Figure 1.** A wireless sensor network with a single actor.

## 1.2. Problem statement

We consider a request-driven large-scale static WSN with IEEE 802.11 carrier sense multiple access with collision avoidance (CSMA/CA) run at medium access control (MAC) layer. Nodes (sensors and actors) are aware of their own position by attached GPS devices or some other positioning algorithm (e.g., [5]). Nodes know the position and other necessary information of their one-hop neighbors by periodic 'hello' message [6]. Each actor spontaneously floods the network with a message carrying its own location; each sensor retransmits (once) only the flooding message from the nearest actor that it has seen. As a result, each sensor knows the globally closest actor.

A sensor reports only to the closest actor upon request (thus request-driven). An actor broadcasts a request message if sensors in the entire network are expected to report or geo-cast [7] to activate only sensors in a necessary subregion. The network is virtually divided into sub-networks, each containing a single actor and the sensors reporting to it. This thus allows us to focus on the single-actor scenario in the sequel without loss of generality. Figure 1(a) depicts a WSN with a single actor. Later, this network topology will be used to exemplify the execution process of our proposed data aggregation protocol.

Sensor reports have the same constant size, which depends on the type of data requested, and are not prioritized. They are treated equally for transmission. A single report may have to be sent using multiple data packets. Data packets are of the same pre-defined size. In this case, reporting delay is roughly linearly proportional, if not equal, to packet delay. For simplicity, we consider that a report is equivalent to a packet and use them interchangeably.

Nodes are equipped with omni-directional antenna of maximum transmission range $R$. They are able to communicate directly if their distance is not larger than $R$. In other words, the network is modeled as a unit disk graph (UDG). A local minimum spanning tree (LMST) [8] sub-graph is spontaneously constructed using one-hop neighborhood information as follows: Each node $u$ computes the Euclidean minimal spanning tree (MST) of the sub-graph $N(u)$ of its one-hop neighbors; an incidental edge $uv$ belongs to LMST if and only if it is in both $MST(N(u))$ and $MST(N(v))$. Figure 1(b) shows an LMST in correspondence to Figure 1(a).

Each edge (communication link) is associated with a cost indicating the minimum transmission power needed for this edge. The cost is computed using the first-order radio model [9]:

$$e(d) = \beta d^{\alpha} + c \tag{1}$$

where $\beta > 0$ is the transmission amplifier, $d$ the transmission distance (edge length), $2 \leq \alpha \leq 6$ a signal attenuation factor, and $c > 0$ the energy for running electronic circuit. For the energy saving purpose, a sensor sends reports along a selected incidental edge using minimum possible power, equal to the associated cost of the edge. Each sensor is aware of its own residual energy.

Given a delay constraint, the goal is to develop a localized energy-efficient (in overall consumption and distribution) data aggregation protocol that achieves high *delay reliability ratio*. The delay reliability ratio of a data aggregation protocol is defined as the ratio of unexpired data packets over all the packets received [4].

### 1.3. Our contributions

We define a novel two-stage delay model based on IEEE 802.11 CSMA/CA MAC layer. The model differentiates low-traffic network scenario (with unsaturated MAC) and high-traffic one (with saturated MAC). In the former case, it represents end-to-end delay using hop count; in the latter case, it expresses delay by degree sum (i.e., the summation of the degree of the nodes along a routing path.) With this delay model, we then design a localized solution, named delay-bounded and energy-efficient data aggregation (DEDA), to the aforementioned data aggregation problem.

In DEDA, data aggregation is triggered by the actor in rounds. At the beginning of each round of data aggregation, a shortest-path (thus energy-efficient) tree is constructed over the LMST, with the actor serving as the root. The tree is built with no concern about delay and serves as the initial data aggregation tree. It will be used without modification if it generates acceptable reporting delay during data aggregation. Otherwise, its structure is changed by replacing LMST edges with selected UDG edges. The judgment and adjustment are done locally at each node on the fly in accordance with a novel concept of *desired progress* (DEP), defined as the ratio of potential delay to remaining lifetime of a report. The final data aggregation tree yields, while respecting the given delay constraint, approximately minimal overall energy consumption and balanced energy usage among sensors.

We further propose to integrate DEDA with a localized sensor activity scheduling algorithm [10] and a localized connected dominating set (CDS) algorithm [11] for energy efficiency improvement. This gives us two variants of DEDA, denoted respectively as A-DEDA and AC-DEDA. Through an extensive set of simulation, we comparatively evaluate the energy efficiency and delay reliability of DEDA (and its two variants) with the only competing protocol MS [4]. Simulation results indicate that DEDA is dramatically more efficient (achieving much longer network lifetime) and has a much higher delay reliability ratio under different network configurations. In particular, they imply that the two-stage delay model is more accurate than the traditional simple hop count-based delay model.

The preliminary version of this work has been published in [12], where only low-traffic scenarios (hop count-based delay model) are considered. The remainder of the article is organized as follows. We briefly review some related work in Section 2. We propose our new delay model in Section 3. We present DEDA in Section 4 and A-DEDA and AC-DEDA in Section 5. Our comparative simulation study is reported in Section 6, followed by the closing remarks drawn in Section 7.

## 2. RELATED WORK

A number of structured data aggregation protocols (e.g., [4,13–18]) have been proposed in literature. Among them, there are not many approaches considering energy efficiency and delay constraint at the same time. We briefly review these previous works in the following sections and introduce the commonly used wireless communication technology IEEE 802.11 CSMA/CA, according to which we will later define our new delay model in Section 3.

### 2.1. Data aggregation

In [13], the authors presented a localized cluster-based aggregation protocol, named LEACH. In this protocol, cluster heads are randomly selected to aggregate collected data from cluster members and transmit them directly to an actor. By in-cluster aggregation, it reduces the amount of information sent over the network, thereby saving significant amount of energy. However, the delay problem was not studied in this work.

In [16], the authors presented a tree-based energy-efficient aggregation protocol. The protocol first computes an LMST sub-graph [8], where transmission energy is considered as the cost of links. Then, it constructs an aggregation tree through a flooding process over the LMST. The aggregation tree is energy efficient, but its construction does not respect any delay constraint.

In [14], the authors studied energy consumption and time delay in data aggregation. They proposed three centralized protocols for constructing an energy sub-optimal aggregation structure. Hop count is used as metric for energy consumption. Although delay was included into consideration, fitting a given specific delay bound was not studied. In [19], the authors presented a centralized protocol to build a delay-bounded MST on the basis of a multicasting algorithm. It uses hop count as energy metric and generates a near-optimal data aggregation tree.

In [15], the authors introduced a structure called delay bound minimum degree spanning tree. Their proposed data aggregation protocol aims at most per-node fairness (in energy consumption) considering delay bound. In the protocol, tree construction requires global information, and hop count is selected as energy metric.

In [17], the authors introduced a scheduling method to make sensors switch among different states and analyzed the energy consumption. The authors in [20] studied the delay bound for data aggregation. But, as opposed to our work here, TDMA MAC layer rather than IEEE 802.11 CSMA/CA is assumed in both articles.

The tradeoffs between energy and latency have been studied in [18]. The proposed approach is to minimize the sum of energy consumption subject to a latency constraint. But, before aggregating data, every sensor might need to wait for its children, which introduces some extra delay.

In [4], the authors presented a distributed multi-state data aggregation framework, referred to as MS in this article, which considers a given delay bound and pursues minimum energy consumption. In MS, a power-aware tree is initially built with no concern about delay and, then, dynamically changed on the basis of measured delay in ongoing traffic. Specifically, the actor maintains a delay reliability ratio and feeds it back to sensors. The tree is locally adjusted if the ratio is higher than a threshold or lower than another threshold. In the former case, each sensor takes as parent the closest neighbor that is closer to the actor than itself for minimizing transmission power. In the latter case, it chooses the neighbor geographically closest to the actor so as to minimize path length and reduce delay.

However, MS has weaknesses compared with our DEDA framework. It consumes more energy and cannot satisfy the delay requirement in dense or high-traffic networks. We will provide detailed comparative analysis in Section 6, when presenting our simulation results.

### 2.2. IEEE 802.11 carrier sense multiple access/collision avoidance

IEEE 802.11 CSMA/CA defines two medium access mechanisms, including distributed coordination function (DCF) and point coordination function. The latter is a centralized scheme and thus off our interest, considering the decentralized nature of WSN. In the following, we briefly introduce the DCF scheme.

At the MAC layer, a node with a packet to transmit monitors the medium status. As soon as the medium is continuously idle for a distributed interframe space (DIFS) slots, it prepares for the transmission. It sets a back-off timer (counting the number of message free slots) randomly selected in a range called *contention window* and transmits after the timer times out. The purpose of having the back-off interval is to reduce collision and to ensure fair medium access. A back-off interval may be broken into pieces because of channel activities. That is, the back-off timer is frozen when the medium is sensed busy and is reactivated after an idle DIFS.

Packet transmission follows a basic access mechanism. Immediately at the end of receiving a packet, the intended receiver waits for a period equal to a short interframe space and signals the sender by an ACK packet. If, within a specified timeout after the packet transmission, the sender does not receive the ACK or senses the transmission of a different packet, it considers the transmission unsuccessful and schedules another transmission attempt following the same algorithm. The size of the contention window is set to a default minimum value $CW_{\min}$ at the first transmission attempt and doubled after each unsuccessful attempt up to a maximum value $CW_{\max}$.

Optionally, a request to send/clear to send (RTS/CTS) access mechanism can be used. Instead of sending the packet immediately after the back-off interval, sender transmits a short frame called RTS. When receiver detects the RTS frame, it responds, after a short interframe space, by a CTS frame. Sender transmits the packet only if the CTS frame is correctly received. Both CTS and RTS frames carry the length of the packet to be transmitted. Hidden nodes overhearing any of the two frames can delay further transmission and avoid collision. This mechanism improves system performance because it reduces the length of the frames involved in the contention process.

Whichever access mechanism is used, we easily have the following conclusion. The more MAC competitors, the higher probability that the medium is busy, and the more often a back-off interval is interrupted, resulting in more frequent waiting, each time for an idle DIFS, and finally larger delay. The relation of MAC delay and number of MAC competitors has been studied under both saturation conditions [21,22] and non-saturation conditions [23,24]. Mathematical models were presented. Despite model difference and complexity, we however observe that MAC delay is approximately a two-stage increasing linear function of number of MAC competitors, as illustrated in Figure 2. The two stages are separated by a point, called a *saturation point*, after which there is at least one packet in the MAC queue whenever a new out-going packet arrives. The saturation point can be determined empirically. With this observation, we propose a novel two-stage delay model in the next section, as opposed to the traditional unthoughtful hop count-based model [4].

## 3. A TWO-STAGE DELAY MODEL

Before proceeding further, we first make the following important definitions to be used throughout the rest of the

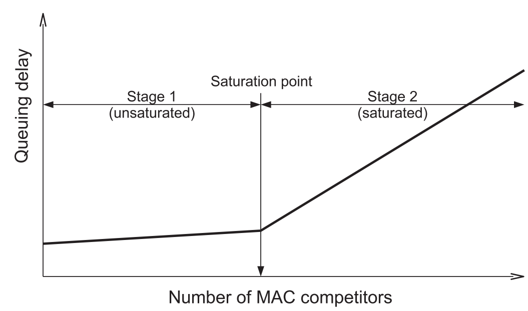



**Figure 2.** Queuing delay versus number of media access control (MAC) competitors.

article, with respect to an arbitrary node $w$ or a source–destination pair $(u, v)$:

- Closed neighborhood $N_w$ of $w$: a node set composed of $w$ and its immediate (one-hop) neighbors.
- Degree $d(w)$ of $w$: the cardinality of node set $N_w$, that is, $|N_w|$.
- Path $P_u^v$ from $u$ to $v$: a forwarder set consisting of $u$ and all the intermediate nodes in order.
- Hop count $h(P_u^v)$ of $P_u^v$: the cardinality of node set $P_u^v$, that is, $|P_u^v|$.
- Degree sum $s(P_u^v)$ of $P_u^v$: the summation of the degree of all nodes in $P_u^v$, that is, $\sum_{w \in P_u^v} d(w)$.

When modeling delay, it is necessary that we understand what it is composed of. As follows, we introduce the delay components.

- Path delay $t(P_u^v)$: the delay that a packet experienced when being transmitted along $P_u^v$; subject to the number of times that the packet is forwarded, that is, $h(P_u^v)$, and the delay of each forwarding.
- Forwarding delay $t(w)$: typically the aggregation of processing delay $t_{\text{proc}}(w)$, queuing delay $t_{\text{queue}}(w)$, transmission delay $t_{\text{tran}}(w)$, and propagation delay $t_{\text{prop}}(w)$ at node $w$ [19].

In forwarding delay $t(w)$, compared with the other delay components, $t_{\text{prop}}(w)$ is negligible and can be considered, if not ignored, as a small constant. As a result of data aggregation, only one report is sent from each sensor in each iteration of data collection. Thus, $t_{\text{proc}}(w)$ and $t_{\text{tran}}(w)$ are approximately constant at different $w$ [25], whereas $t_{\text{queue}}(w)$ is dependent on the MAC layer used and can be rather different. For a single report, at each node $w$, we define *delay constant* as $c = t_{\text{proc}}(w) + t_{\text{prop}}(w) + t_{\text{tran}}(w)$. Forwarding delay $t(w)$ and path delay $t(P_u^v)$ are given by

$$t(w) = t_{\text{queue}}(w) + c \tag{2}$$

$$t(P_u^v) = \sum_{w \in P_u^v} t(w) \tag{3}$$

Let $\alpha$ be a constant *delay factor*. We define the *delay variable* of $w$ and $P_u^v$ respectively as

$$\upsilon(w) = \frac{1}{\alpha} t_{\text{queue}}(w) + \frac{c}{\alpha} \tag{4}$$

$$\upsilon(P_u^v) = \sum_{w \in P_u^v} \upsilon(w) \tag{5}$$

Equations (2) and (3) can then be rewritten as follows:

$$t(w) = \alpha \times \upsilon(w) \tag{6}$$

$$t(P_u^v) = \alpha \times \upsilon(P_u^v) \tag{7}$$

As a convention, we measure the *distance* between two nodes $u$ and $v$ in a given network graph by the length of a 'shortest' path connecting them in the graph. Denote the length of the hop led by node $w$ by $l(w)$ and the length of path $P_u^v$ by $l(P_u^v)$. Intuitively, $l(P_u^v)$ should be a function of path delay variable $\upsilon(P_u^v)$ so that it reflects the delay that a packet would experience when traveling along $P_u^v$.

We define that traffic intensity is low if each node has a unsaturated channel (light MAC competition) and high otherwise. According to Figure 2, in a low-traffic scenario, $t_{\text{queue}}(w)$ is constant (roughly). Therefore, $\upsilon(w)$ has a constant value, and $\upsilon(P_u^v)$ is proportional to hop count $h(P_u^v)$. In a high-traffic scenario, $t_{\text{queue}}(w)$, and thus $\upsilon(w)$, is proportional to $d(w)$, rendering $\upsilon(P_u^v)$ proportional to degree sum $s(P_u^v)$.

$$\upsilon(w) \propto \begin{cases} 1 & \text{(low traffic)} \\ d(w) & \text{(high traffic)} \end{cases} \tag{8}$$

Hence, in these two scenarios, we are allowed to consider $l(w)$ and $l(P_u^v)$, respectively as

$$l(w) \equiv \begin{cases} 1 & \text{(low traffic)} \\ d(w) & \text{(high traffic)} \end{cases} \tag{9}$$

$$l(P_u^v) \equiv \begin{cases} h(P_u^v) & \text{(low traffic)} \\ s(P_u^v) & \text{(high traffic)} \end{cases} \tag{10}$$

In order words, the distance between $u$ and $v$ is the length of the path connecting $u$ and $v$ with the smallest hop count in low-traffic network scenarios or the smallest degree sum in high-traffic network scenarios. The delay mode (Equations (6) and (7)) is correspondingly referred to as *hop count based* or *degree sum based*.

## 4. THE DELAY-BOUNDED AND ENERGY-EFFICIENT DATA AGGREGATION SCHEME

We shall now present our new data aggregation framework DEDA. During the presentation, we will use the network topology in Figure 1 as an example to illustrate its execution. DEDA builds a delay-bounded and energy-efficient data aggregation tree on the basis of a novel concept of DEP, using edges selected from LMST and UDG. The term 'progress' means reduction in distance to the actor. The measurement of distance is subject to traffic intensity and needs to be determined in advance. By using different distance measurements, we will obtain two DEDA versions: hop count-based DEDA (with hop count-based delay model) and degree sum-based DEDA (with degree sum-based delay model).

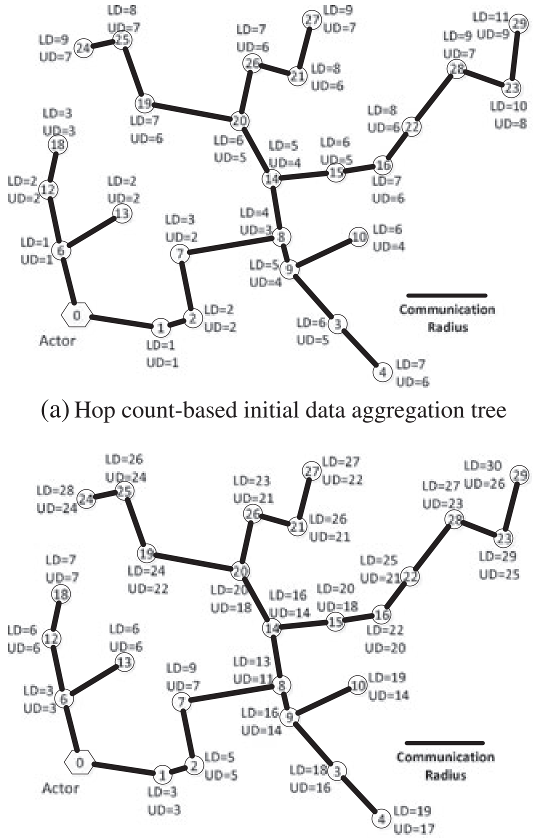
### 4.1. Building initial data aggregation tree

The initial data aggregation tree may be constructed at different time. When to construct is indeed user's choice. The tree can be computed only once, at the beginning of the network operation period when the actor floods the network to advertise its location. It is then locally stored and adjusted in each round of data aggregation. In this way, the tree will however be vulnerable to node failures. Alternatively, it can be repeatedly computed at the beginning of each data aggregation round, upon the actor's data aggregation request. By this means, fault tolerance ability improves while message cost obviously increases. Here, we choose dynamic construction for DEDA.

When necessary (upon user's query request), the actor floods the entire network with a request message using maximum transmission power. The message includes a sequence number $SN$, a reporting delay limit $DL$, the UDG distance $UD$ and LMST distance $LD$ to the actor, and sender $id$. The sequence number $SN$ is a monotonically increasing number. It is incremented by the actor every time when it initiates a new request. Recall that distance is defined to reflect delay by considering the packet delay per unit distance (or transmission speed) as constant. The delay limit $DL$ is given by the end user and is application-dependent (and same for all sensors). It is measured in distance. This implies that each report should reach the actor within this distance in order to remain useful. These two fields remain constant. The other three fields $LD$, $UD$, and $id$ are initially set to 0 or $null$, and are to be updated by each forwarding sensor.

In order to identify fresh or stale requests, each sensor records the largest $SN$ (and the corresponding $DL$) that it has seen so far. Stale requests are discarded. When a sensor $w$ receives a new request for the first time, it sets its own UDG distance (to the actor) to $UD + l(w)$ and it will set its LMST distance to $LD + l(w)$ if the sender is an LMST neighbor, where $UD$ and $LD$ are obtained from the request message. In the case that it receives the same fresh request again, it updates the two distances only if their current values are larger. Node $w$ retransmits the received request message if and only if the message has caused the reset or update of its UDG distance or LMST distance. For each retransmitted message, it remembers the sender; before retransmission, it updates the message with its own $id$, $UD$, and $LD$. Retransmission is carried out using maximum transmission power.

After this flooding process, an LMST-based shortest-path tree is established as shown in Figure 3. This tree is rooted at the actor. Every sensor now knows its UDG distance, LMST distance, and its parent (identified by stored sender $id$). It also learns about all parent (child) candidates, which are the neighbors with equal or smaller (resp., larger) LMST distance. The tree is an initial data aggregation tree, along which sensors aggregate and send reports upward to the actor. It is subject to change during data aggregation when delay limit is taken into consideration, as discussed in the next subsection.



(a) Hop count-based initial data aggregation tree

(b) Degree sum-based initial data aggregation tree

**Figure 3.** Initial data aggregation tree in delay-bounded and energy-efficient data aggregation.

## 4.2. Taking delay limit into consideration

**Definition 1** (Expectedness)**.** *Consider a sensor at UDG distance k from the actor. Suppose that the sensor receives a report with already experienced delay m. The report is* expected *if* $m + k \leq DL$*, and* unexpected *otherwise.*

**Definition 2** (Desired progress)**.** *The DEsired Progress (DEP) of a sensor is defined as the desired LMST distance progress led by w toward the actor. It is computed by* $DEP = \left\lceil \frac{LD(w)}{DL-MED} \right\rceil \times l(w)$*, where* MED *is the most experienced delay of all expected reports (*$MED = 0$*, if no expected report) and* $l(w)$ *is a constant variable.*

The LMST-based shortest-path tree is a data aggregation structure with approximately minimal energy cost (because of the cost definition of each edge). However, it may not satisfy the delay requirement. Indeed, a report can be delivered along the tree to the actor by the specified deadline if the report progresses an LMST distance of $DEP$ at each intermediate node on average. Otherwise, sensors must seek shortcuts in UDG that can provide larger LMST-distance progress. This involves use of non-LMST edges and will change the tree structure, as we show in the following subsection.


*Wirel. Commun. Mob. Comput.* 2011; **11**:1603–1617 © 2011 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm

### 4.3. Selecting shortcut for fitting delay limit

When use of the LMST-based shortest-path tree does not meet the delay requirement, DEDA seeks shortcuts in UDG according to the *DEP* value computed at each node. This results in an approximately balanced data aggregation tree with height roughly equal to the delay limit $DL$. We will now describe when and how data aggregation shortcuts are locally selected. The pseudo-codes are given in Algorithm 1.

If sensor $w$ has an empty child candidate set, it starts the aggregation process immediately after receiving the request from the actor. The sensor computes its *DEP* value and selects as parent the one (among all parent candidates) that leads to a progress (in LMST) equal to this value. In the presence of multiple such candidates, the one within the shortest Euclidean distance is chosen so as to save transmission power for reporting. If *DEP* cannot be satisfied exactly, the parent candidate whose progress is closest to it is taken, with preference given to the one with the greatest progress in case of tie. Finally, any remaining tie can be broken by node $id$. Once $w$ decides its parent, it declares this decision at maximum transmission power. The decision message includes sender $id$, parent node $id$, and the evaluated reporting delay $ERD = l(w)$ at $w$. Other information can be added, as per query.

Algorithm 1: Shortcut selection (at node $w$)

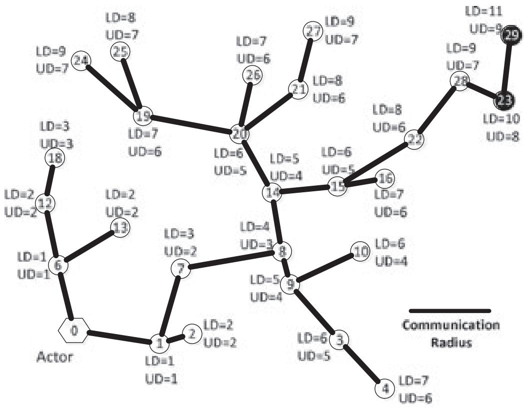
```
On receiving request message REQT
if REQT.SN > SN
    SN := REQT.SN
    DL := REQT.DL
    UD := REQT.UD + l(w)
    if REQT.SENDER is LMST neighbor
        LD := REQT.LD + l(w)
    endif
    update and retransmit REQT message
endif
if REQT.SN = SN
    if UD > REQT.UD + l(w)
        UD := REQT.UD + l(w)
    endif
    if (REQT.SENDER is LMST neighbor)
          && (LD > REQT.LD + l(w))
        LD := REQT.LD + l(w)
    endif
    if UD or LD is updated
        update and retransmit REQT message
    endif
endif
if received REQT with latest SN from all neighbors
    if no child candidates
        compute DEP value by Definition 2
        select PARENT according to DEP
        broadcast PARED message with ERD = l(w)
    endif
endif

On receiving parent decision message PARED
if (w = PARED.PARENT) && (PARED.ERD <= DL - UD)
    CHILD := CHILD union {PARED.SENDER}
    if PARED.ERD > MED
        MED := PARED.ERD
    endif
endif
if having received PARED from all child candidates
    compute DEP value by Definition 2
    select PARENT according to DEP
    broadcast PARED message with ERD = MED + l(w)
endif

On receiving a report
if the report is expected
    buffer the report locally
endif
if all buffered reports are ready
    aggregate the reports into a single report
    send the aggregated report to PARENT
endif
```

If sensor $w$ has a non-empty child candidate set, it will wait for parent declarations from the child candidates. For each received decision message, $w$ deletes the sender from the candidate list, and it will add the sender to the child list if it is the selected parent of the sender. For each child, $w$ considers the *ERD* in the corresponding parent declaration as experienced delay of future reports from the child. According to this value, it evaluates whether those reports will be expected and then marks the child to be expected or unexpected accordingly. If the child is an expected child and the *ERD* is larger than the locally maintained *MED*, $w$ will update *MED* to *ERD*. When the child candidate list becomes empty, $S$ starts parent selection and broadcasts the result by decision message, where the *ERD* field is set to $MED + l(w)$.

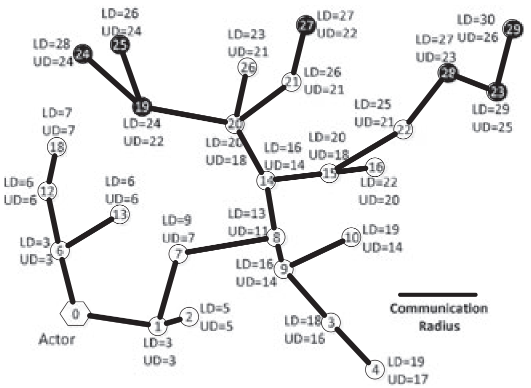
In correspondence to Figure 3, the final data aggregation tree is constructed in Figure 4. Examine sensor **22** whose original parent is node **16**. In a low-traffic network scenario with $DL = 7$ (where the hop count-based delay model is applied), it receives a parent decision message from child candidate **28** with $ERD = 1$ and then decides whether **28** is expected or not. Because $ERD(\mathbf{28}) + UD(\mathbf{22}) = DL$, **28** is expected to **22**, and any report received from **28** will therefore be forwarded by **22**. Node **22** finally changes its parent to node **15** as **15** leads to a progress rate $(LD(\mathbf{22}) - LD(\mathbf{15}) = 2)$ equal to its $DEP = \lceil \frac{8}{7-1} \rceil \times 1 = 2$. In a high-traffic network scenario with $DL = 21$ (where the degree sum-based delay model is applied), **22** receives a parent decision message from **28** with $ERD = 2$. Because $ERD(\mathbf{28}) + UD(\mathbf{22}) > DL$, it marks **28** as unexpected and ignores any report from **28** in the future. Afterwards, **22** changes its parent to **15** because **15** leads to a progress $(LD(\mathbf{22}) - LD(\mathbf{15}) = 5)$ closest to its $DEP = \lceil \frac{25}{21-0} \rceil \times d(\mathbf{22}) = 6$ among all parent candidates.

## 5. FURTHER IMPROVEMENT

In this section, we propose two DEDA variants, A-DEDA and AC-DEDA, which require less sensors to report and thus achieve improved performance on both energy efficiency and delay reliability.




(a) Hop count-based final data aggregation tree (DL = 7)




(b) Degree sum-based final data aggregation tree (DL = 21)

**Figure 4.** Final data aggregation tree in delay-bounded and energy-efficient data aggregation.

A-DEDA adopts a localized sensor activity scheduling algorithm [10] for selecting an active node set. Active nodes monitor the environment and generate reports, whereas the others switch to sleep mode for energy saving. DEDA is therefore run only on active nodes. In [10], each node sets a timeout $t$ to start coverage evaluation and schedule its activity. Considering that nodes with smaller $t$ will have a higher chance to stay active, $t$ is set to be inversely proportional to nodal remaining energy level. This definition favors nodes with more residual energy. AC-DEDA is a combination of A-DEDA and a localized CDS construction algorithm [11], which is run on the active nodes determined by [10]. Each active node either belongs to the CDS or has a direct active neighbor in it. Non-CDS active nodes report to the closest CDS neighbors. CDS nodes run DEDA for data aggregation.

Figure 5 demonstrates A-DEDA in a low-traffic network. For illustrative purpose, low-$id$ nodes are considered to have more residual energy than high-$id$ nodes. By the activity scheduling algorithm, five nodes (with a white dot) are scheduled to sleep. The rest stay active and run DEDA. An initial data aggregation tree (see Figure 5(b)) is built over the LMST (Figure 5(a)) of these active nodes. All active nodes adjust their parent selection according to the *DEP* values that they locally calculate. The final data aggregation tree is shown in Figure 5(c).

Figure 6 demonstrates AC-DEDA in the same network configured as in Figure 5. Low-$id$ nodes are given a high priority to join CDS. Before data aggregation, the status of every node could be one of the following: CDS node, non-CDS node (active but not in CDS), or passive node. The actor floods a request along the LMST of CDS nodes (Figure 6(a)). After the flooding process, an initial data aggregation tree is built as in Figure 6(b). Non-CDS nodes are attached to the tree via their closest CDS neighbors. CDS nodes implement DEDA to determine their position in the aggregation tree. The final data aggregation tree is shown in Figure 6(c).

## 6. PERFORMANCE EVALUATION

We evaluate DEDA family protocols in comparison with the only known delay-bounded energy-aware data aggregation protocol MS [4] through extensive simulation. We use per-node energy consumption, network lifetime (the time when the first sensor depletes its battery power) and delay reliability ratio as the performance metrics. Without loss of fairness, we omit the energy consumption in aggregation processing and idle state, which are identical for the tested protocols. As we will see, our simulation results indicate that DEDA family protocols are indeed more delay reliable and more energy efficient than MS, and our expectation about the improved performance (both energy efficiency and delay reliability) of A-DEDA and AC-DEDA is also confirmed.

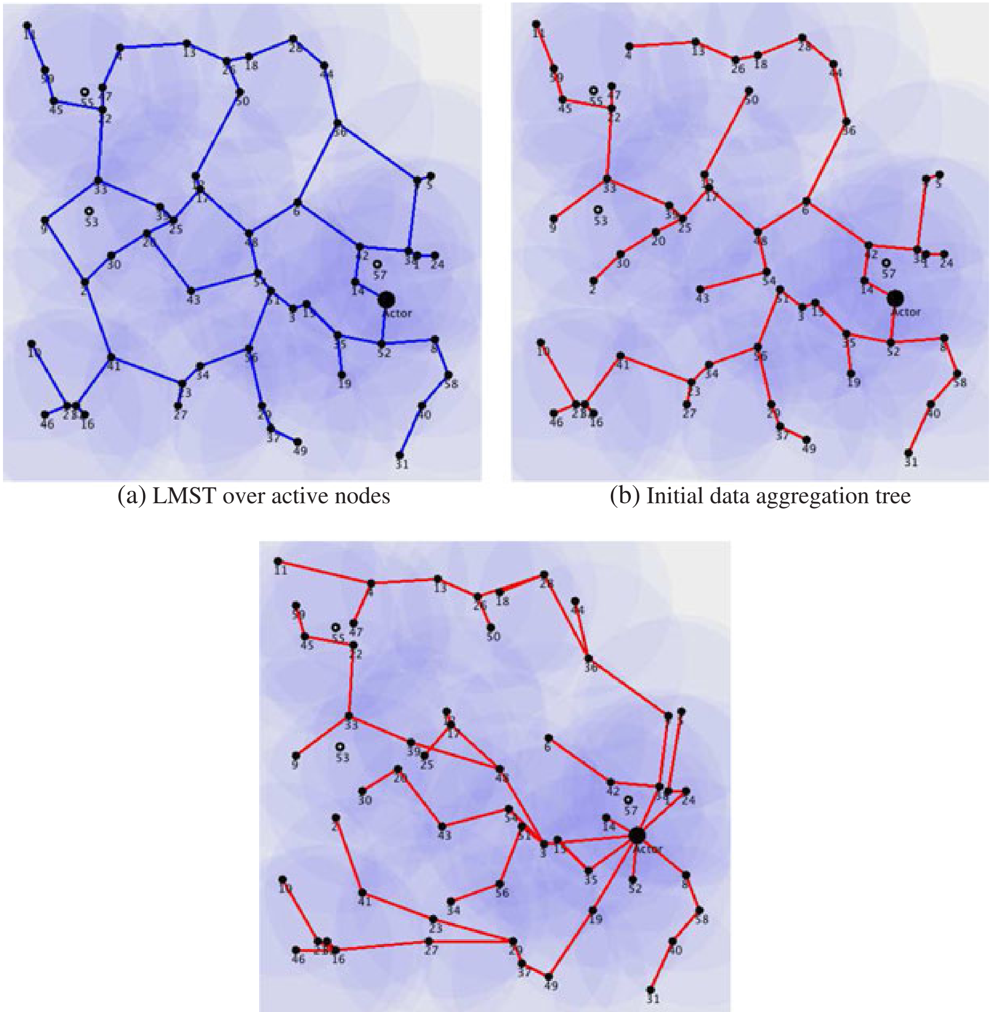
### 6.1. Simulation setup

We implemented DEDA and MS within the Glomosim network simulator [26]. In our simulation, we used a static network of 60 sensors and a single actor, with 802.11 CSMA/CA employed at MAC layer. The ratio of communication range to sensing range was set to 2. Sensors were configured to initially have the same amount of energy. The first-order radio model $e = \beta d^{\alpha} + c$ was adopted as the energy model for transmission, where $\beta = 100$ pJ/bit/m$^{\alpha}$ and $\alpha = 4$. All data packets had the same constant size of 200 bits.

The protocol MS uses three routing sub-routines, start-up, greedy, and aggregation, respectively for three working states. It relies on a state switching scheme to control sensor's routing sub-routine selection. In the switching scheme, there are several parameters to be configured:

- High event reliability threshold $r_{\text{th}}{}^{+} = r_{\text{th}} + \varepsilon$;
- Low event reliability threshold $r_{\text{th}}{}^{-} = r_{\text{th}} - \varepsilon$;
- State switching probability $P_{\text{switching}}$.

The actor computes and releases feedbacks after receiving every data packet to inform sensors to make working state




(a) LMST over active nodes (b) Initial data aggregation tree

(c) Final data aggregation tree

**Figure 5.** A-DEDA in a low-traffic network. LMST, local minimum spanning tree.

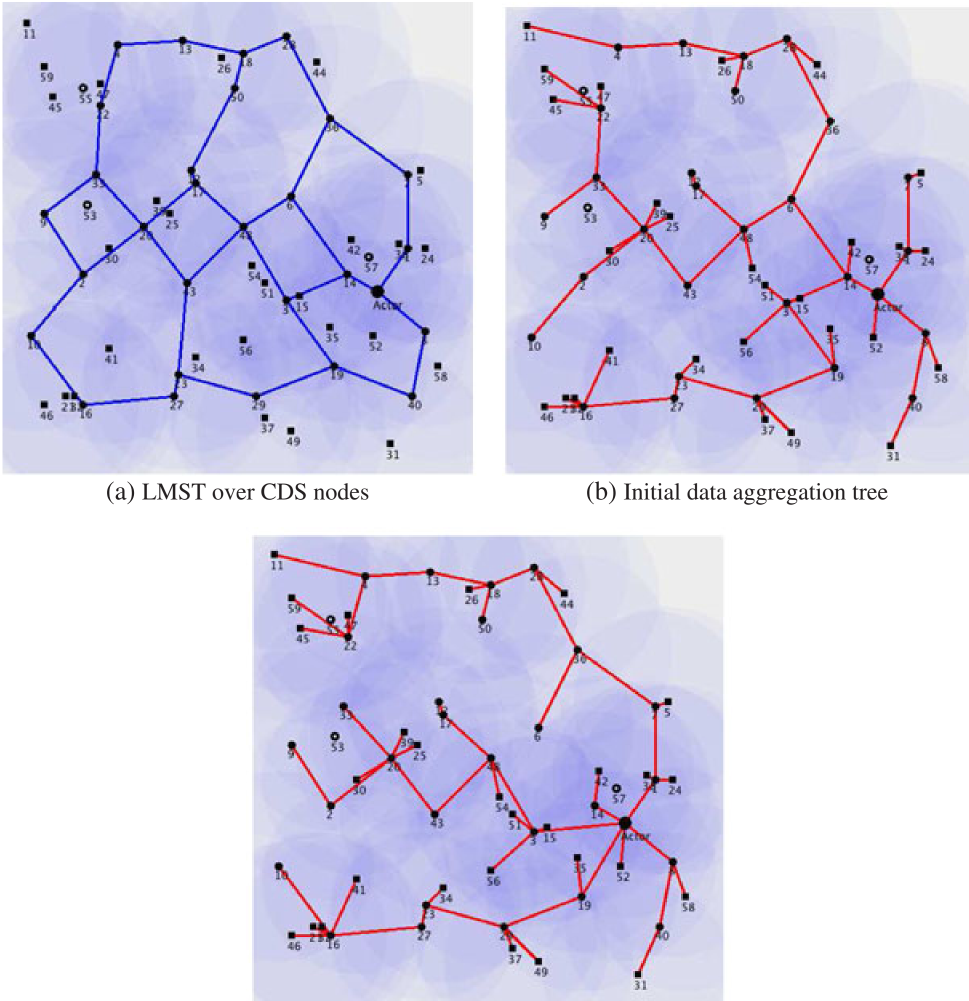
decision. In our simulation, we set $\varepsilon = 3\%$ and $P_{\text{switching}} = 70\%$. We set $r_{\text{th}}$ to the delay reliability ratio of DEDA under the same network configuration so that we can compare the energy efficiency of MS and DEDA.

Because sensors are transmitting simultaneously, neighboring sensors are MAC competitors. We are thus allowed to control traffic intensity by changing the average node degree $d$, which is in turn controlled by adjusting nodal communication range. Through simulation experiments, we identified that the MAC saturation point is $d = 8$. The average MAC delay $t_{\text{queue}}$ is about 6500 $\mu$s (simulated time) for $d \le 8$ (i.e., low-traffic scenarios) and roughly $1000 * d$ $\mu$s for $d \ge 8$ (i.e., high-traffic scenarios). Recall the definition of distance in Section 3. Thus, given $d$, we are able to convert a delay limit $T_{\max}$ expressed in time to a delay limit $DL$ in distance by $DL = T_{\max}/t_{\text{queue}}$.

We evaluated DEDA family protocols and MS both in a low-traffic scenario and in a high-traffic scenario. Low traffic is obtained by setting $d = 6$, whereas high traffic is accomplished by letting $d = 14$. DEDA family protocols engages the hop count-based delay model in the former case and the degree sum-based model in the latter case, whereas MS always used the hop count-based model. We employed two delay limit settings $DL = 4, 9$, and varied the electronic unit energy consumption $c$ in the energy model (see Equation (1)) and the ROI size $l^2$ for energy efficiency study. We additionally cross-verified the two delay models with different network density $d$ (reflecting traffic intensity), by fixing $DL = 10$ and studying the delay reliability ratio of DEDA family protocols in relation with $d$. For each simulation setting, we conducted 50 simulation runs with randomly generated network graphs and took the average results.

## 6.2. Simulation results

In MS, each sensor selects its next hop on the basis of a two-hop rule when building the initial data aggregation



(a) LMST over CDS nodes

(b) Initial data aggregation tree

(c) Final data aggregation tree

**Figure 6.** AC-DEDA in a low-traffic network. LMST, local minimum spanning tree; CDS, connected dominating set.

tree. Consider an arbitrary sensor $w$. Denote by $a$ the actor and by $u(\cdot)$ the utility function for hop selection. Suppose that $w$ selects a neighboring node $n$ for which $u(|wn|) + u(|na|)$ is minimized over all neighbors. Let $y = |wn|$ and $x = |na|$. Consider another neighbor $b$ of $w$ so that $y + \Delta = |wb|$ and $x + \nabla = |ba|$. It can be shown that the difference between the two neighbors in considered criterion is $u(|wn|)+u(|na|)-u(|wb|)-u(|ba|) \approx \alpha y^{\alpha-1}\Delta + \alpha x^{\alpha-1}\nabla$. When $w$ is far from $a$, we have $y \ll x$ and the difference is $\approx \alpha x^{\alpha-1}\nabla$. This means that the selection is biased toward the neighbor closest to $a$; that is, the selection is close to the one made in greedy routing. This is not really a power efficient selection but a solution minimizing hop count. In DEDA, the initial data aggregation tree is a shortest-path tree based on the transmission power rather than hop count, thus efficient in overall energy consumption.

In MS, simple and inflexible greedy principle is engaged to adjust the initial data aggregation tree. Each node repeatedly selects the same neighbor as parent, making individual nodes' energy usage unbalanced. The greedy strategy for minimizing transmission power at each node yields long data aggregation paths with short hops. Such a path is not efficient in overall energy consumption unless the electric unit energy consumption constant $c$ in the energy model (see Equation (1)) is equal to 0, which we know is impossible in reality. DEDA does not have these problems because its tree adjustment method is based on the moderate DEP concept rather than the extreme greedy rule. Summarizing, in MS, the use of inefficient initial tree and greedy tree adjustment method induces unoptimized and unbalanced energy consumption, rendering the protocol less helpful for energy saving and network lifetime elongation than DEDA. The difference in their performance is expected to be large with loose delay requirement and high electric unit energy consumption, that is, large-valued $DL$ and $c$, and to be small with strict delay requirement and low electric unit energy consumption, that is, small-valued $DL$ and $c$. The aforementioned analysis is confirmed by our simulation results to be presented later. Note that the energy consumption results are obtained right after the fifth round of data aggregation.

### 6.2.1. Impact of delay bound.

Observe Figures 7–9. As the delay requirement becomes loose, that is, when $DL$ increases, the tested protocols all perform increasingly better in energy efficiency. This is because they are given more room to make better hop selection energy-wise. This phenomenon is very intuitive and easy to understand.

### 6.2.2. Impact of electric unit energy consumption.

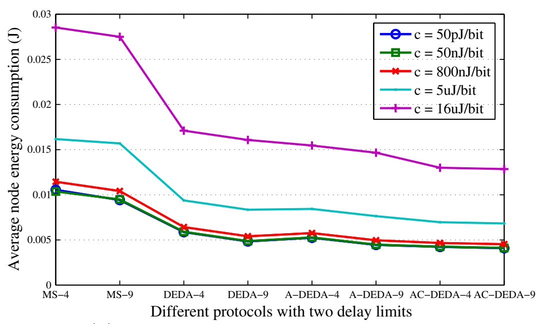
This parameter does not have any impact on the delay reliability of the tested protocols, where hop selection is made in a localized way without global knowledge. We focus only on its impact on the energy efficiency of the protocols.

We set $l = 100$ m and varied $c$ in the range between 50 pJ/bit and 6 µJ/bit. The simulation results are plotted in Figure 7. Observe that both DEDA family protocols and MS have degrading performance as $c$ increases. This is because the effectiveness of increasing hop count (i.e., having short hops) for energy saving is increasingly weakened by the growth of aggregated electric unit energy consumption. But nevertheless, DEDA family protocols perform significantly better than MS. As expected, A-DEDA and AC-DEDA are more efficient than DEDA, especially in dense networks. Compared with DEDA, they respectively save 10% and 25% energy in sparse networks, and 50% and 75% $\sim$ 80% energy in dense networks.

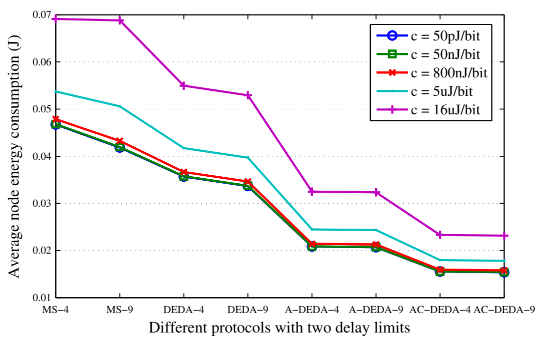
### 6.2.3. Impact of region-of-interest size.

When the network size is fixed, change of ROI size leads to change of $d$ (i.e., network density), given that the network connectivity must be preserved. The impact of ROI size on the delay reliability ratio can be evaluated equivalently by varying network density. Here, we focus on its impact on energy efficiency.

We fixed $c = 50$ pJ/bit and varied $l$ among 50, 100, 150, 200, and 250 m. With random node distribution, the larger the ROI, the longer communication link on average (for connectivity preservation), and therefore, the more energy is consumed for transmission. Our simulation results in Figure 8 indicate this phenomenon clearly. We observe that DEDA family protocols exhibit dramatically better performance (up to 50% energy saving) than MS in ROI of different sizes. Compared with DEDA, A-DEDA and AC-DEDA are more efficient thanks to their reduced sensor

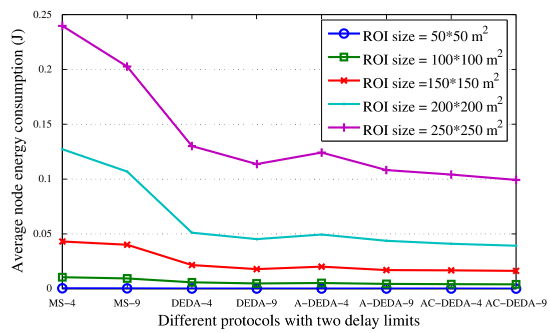



(a) Hop count-based delay model ($d = 6$)

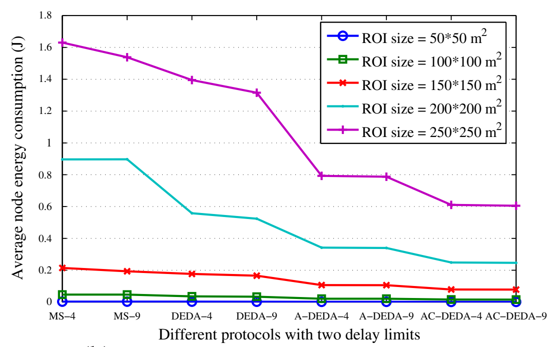



(b) Degree sum-based delay model ($d = 14$)

**Figure 7.** Per-node energy usage with different $c$ and $DL$. DEDA, delay-bounded and energy-efficient data aggregation; MS, distributed multi-state data aggregation.

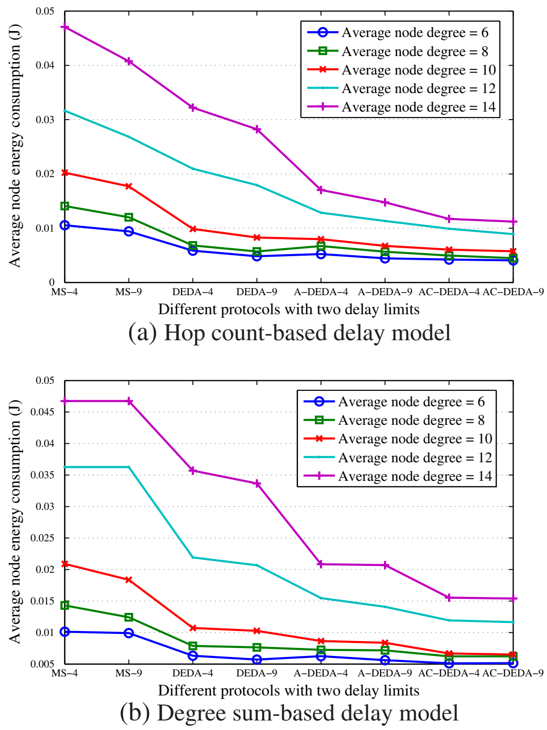



(a) Hop count-based delay model ($d = 6$)




(b) Degree sum-based delay model ($d = 14$)

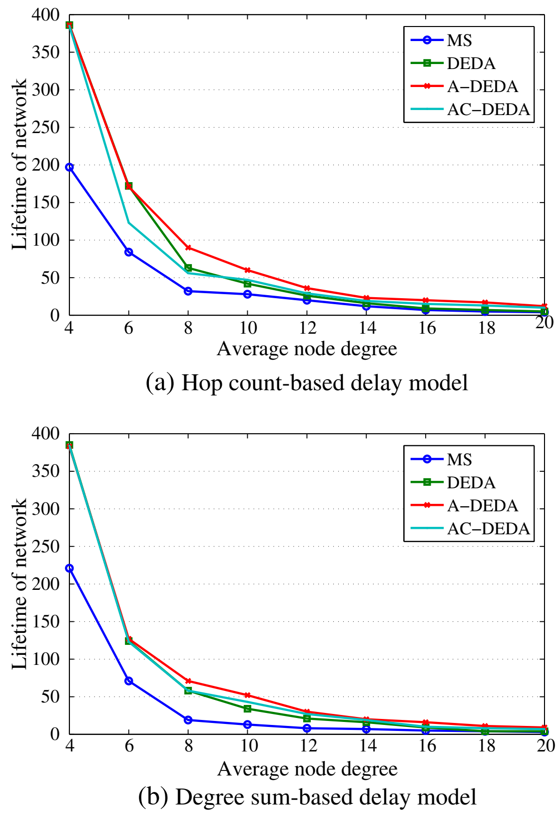
**Figure 8.** Per-node energy usage with different $l$ and $DL$. DEDA, delay-bounded and energy-efficient data aggregation; MS, distributed multi-state data aggregation; ROI, region of interest.




(a) Hop count-based delay model

(b) Degree sum-based delay model

**Figure 9.** Per-node energy usage with different $d$ and $DL$. DEDA, delay-bounded and energy-efficient data aggregation; MS, distributed multi-state data aggregation.




(a) Hop count-based delay model

(b) Degree sum-based delay model

**Figure 10.** Network lifetime with different $d$ ($DL = 10$). DEDA, delay-bounded and energy-efficient data aggregation; MS, distributed multi-state data aggregation.

activities and network traffic. In particular, the superiority is obvious in dense networks deployed in a large ROIs. This is because, in our simulation, such networks have to use very long communication links in order to satisfy the degree requirement, which offers great opportunity for A-DEDA (AC-DEDA) to save energy by activity scheduling (and CDS formation).

### 6.2.4. Impact of network density.

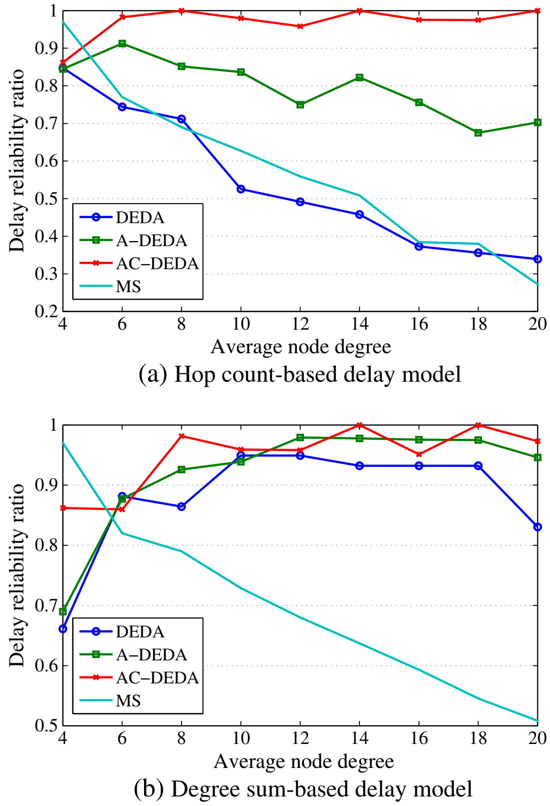
In the sequel, we will study how network density impact the energy efficiency and delay reliability of the tested protocols. For this study, we fixed $c = 50$ pJ/bit and $l = 100$ m, and varied $d$ in the range between 4 and 20.

An increased $d$ is implemented by enlarging nodal communication range, which implies that the average hop length increases as $d$ goes up and the amount of energy consumed by each node grows as a result. Hence, DEDA family protocols and MS show an ascending trend in energy consumption with $d$ in Figure 9. DEDA saves 30%–50% energy over MS in general. Compared with DEDA, A-DEDA and AC-DEDA respectively save 2%–50% and 60%–70% when the hop count-based delay model is applied, and 10%–45% and 10%–30% when the degree sum-based delay model is engaged. Figure 10 shows that DEDA family protocols are able to achieve more balanced energy consumption among sensors and thus longer network lifetime (possibly > 100% longer) than MS. We notice that the degree sum-based versions of DEDA may consume a little bit more energy (about 5%–10%) than the hop count-based counterparts when $d$ is large. The reason is that the degree sum-based delay model reflects the real delay situation more accurately in high-traffic networks, and the protocol sometimes has to select energy-inefficient links to satisfy the delay limit.

The relation of delay reliability ratio and node degree is shown in Figure 11 with $DL = 10$. We examine the performance of the protocols when the hop count-based delay mode is employed in Figure 11(a). The ratio is about 85% in the case of $d = 4$ for all variants of DEDA because, in such sparse networks, the improvement by activity scheduling and CDS is negligible. We see that in DEDA, it quickly decreases to 35% or so when $d$ climbs to 20, where A-DEDA is still able to achieve fairly high delay reliability ratio, approximately 70%, and AC-DEDA even gets nearly 100% reliability ratio. This superior performance is due to the activity scheduling algorithm and CDS algorithm that reduce network traffic and thus MAC delay at individual nodes. MS has nearly the same performance as DEDA because it indeed takes the delay reliability ratio of DEDA as the expected delay reliability ratio.

In Figure 11(b), we observe the protocols performance when the degree sum-based delay mode is adopted. In the


*Wirel. Commun. Mob. Comput.* 2011; **11**:1603–1617 © 2011 John Wiley & Sons, Ltd.
DOI: 10.1002/wcm




(a) Hop count-based delay model

(b) Degree sum-based delay model

**Figure 11.** Delay reliability ratio with different $d$ ($DL = 10$). DEDA, delay-bounded and energy-efficient data aggregation; MS, distributed multi-state data aggregation.

case of $d \geq 8$, DEDA has a fairly stable delay reliability ratio, larger than 90%. Its performance drops dramatically when $d < 8$ and can be worse than the hop-based versions. MS is input with 95% as expected delay reliability ratio, which is the average delay reliability ratio of DEDA family protocols for $d \geq 8$. We find that MS, however, produces this ratio only when $d < 8$. Its delay reliability drops rapidly in a linear fashion for $d > 8$ and reaches 50% when $d = 20$, while it consumes significantly more energy than DEDA family protocols (see Figure 11(b)). The results imply that our proposed two-stage delay model is reasonable and well captures the delay behavior of packet transmission and that DEDA family protocols are superior to MS from algorithmic point of view.

## 7. CONCLUSIONS

In this article, we have addressed the data aggregation problem with an emphasis on delay bound and energy efficiency. We proposed a novel two-stage delay model based on IEEE 802.11 CSMA/CA MAC layer. This model uses hop count to measure end-to-end delay when the network traffic is low (unsaturated channel), and degree sum along the routing path when the traffic is high (saturated channel). We then introduced the novel concept of DEP for hop selection and devised a localized DEDA protocol accordingly.

To balance nodes energy consumption and reduce overall network energy usage, DEDA first builds an energy optimal aggregation tree and then fixes it according to the DEP value computed at each node so that it does not violate the given delay bound. We also presented two DEDA variants, A-DEDA and AC-DEDA, by combining it with a sensor activity scheduling algorithm [10] and a localized CDS algorithm [11]. Compared with DEDA, the two variants have increased energy efficiency and delay reliability. Simulation results showed that DEDA family protocols are much more energy efficient in overall consumption and distribution than the only known competing protocol MS [4], and our proposed delay model is more reasonable and, when used, leads to significantly better delay reliability than traditional simple hop count-based model.

Delay-bounded and energy-efficient data aggregation requires the knowledge of average node degree so as to determine the delay model (distance measurement) to be used. This information can be readily computed by the actor if sensors attach their degree information to data packets. The actor then embeds the information in the data aggregation requests and passes to every sensor. Sensors may fail or enter sleeping mode, breaking the established data aggregation tree. In the former case, neighboring sensors will detect the failure and notify the actor to reinitiate data aggregation and reconstruct the data aggregation tree. In the latter case, the sleeping sensors themselves will inform the actor before switching off.

Delay-bounded and energy-efficient data aggregation can be extended to solve data gathering (data collection without aggregation) problem. Because data from different source nodes have to be sent separately, the LMST-based shortest-path tree will not be an energy optimal collecting structure anymore. Consequently, DEDA and its variants proposed in this article will not be as energy efficient as they are for data aggregation. In such applications, every node should find energy and delay costs along the path to the actor with minimal overall energy costs and send the information to its neighbors. According to this knowledge of neighbors, each node will be able to compute its DEP value in report transmission.

## ACKNOWLEDGEMENTS

This work was partially supported by NSERC Strategic Grants STPSC356913-2007B and STPGP 336406-07, and NSERC Collaborative Research and Development Project CRDPJ 386874-09.

## REFERENCES

1. Chen J, Xu W, He S, Sun Y, Thulasiramanz P, Shen X. Utility-based asynchronous flow control algorithm for

wireless sensor networks. *IEEE Journal on Selected Areas in Communications* 2010; **28**(7): 1116–1126.

2. He S, Chen J, Yau DKY, Sun Y. Cross-layer optimization of correlated data gathering in wireless sensor networks. *IEEE Transactions on Mobile Computing*. DOI: 10.1109/TMC.2011.210.
3. Zhang J, Shen X, Dai G, Feng Y, Tang S, Lv C. Energy-efficient lossy data aggregation in wireless sensor networks. *Ad Hoc & Sensor Wireless Networks* 2011; **11**(1-2): 111–135.
4. Melodia T, Pompili D, Gungor VC, Akyildiz IF. Communication and coordination in wireless sensor and actor networks. *IEEE Transactions on Mobile Computing* 2007; **6**(10): 1116–1129.
5. Li X, Mitton N, Simplot-Ryl I, Simplot-Ryl D. Dynamic beacon mobility scheduling for sensor localization. *IEEE Transactions on Parallel and Distributed Systems* 2012. To appear.
6. Li X, Mitton N, Simplot-Ryl D. Mobility prediction based neighborhood discovery for mobile ad hoc networks, In *Proceedings of IFIP NETWORKING*, 2011; 241–253.
7. Stojmenovic I. Geocasting with guaranteed delivery in sensor networks. *IEEE Wireless Communications Magazine* 2004; **11**(6): 29–37.
8. Li N, Hou JC, Sha L. Design and analysis of an MST-based topology control algorithm. *IEEE Transaction on Wireless Communications* 2005; **4**(3): 1195–1206.
9. Rodoplu V, Meng TH. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications* 1999; **17**(8): 1333–1344.
10. Gallais A, Carle J, Simplot-Ryl D, Stojmenovic I. Localized sensor area coverage with low communication overhead, In *Proceedings of IEEE PerCom*, 2006; 328–337.
11. Carle J, Simplot-Ryl D. Energy efficient area monitoring by sensor networks. *IEEE Computer Magazine* 2004; **37**(2): 40–46.
12. Xu C, Li X, Nayak A, Stojmenovic I. Localized delay-bounded and energy-efficient data aggregation in low-traffic request-driven wireless sensor and actor networks, In *Proceedings of IWCMC*, 2011.
13. Heinzelman W, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks, In *Proceedings of Hawaii International Conference on System Sciences*, 2000.
14. Krishnamachari B, Estrin D, Wicker S. The impact of data aggregation in wireless sensor networks, In *Proceedings of Distributed Event-Based Systems(DEBS)*, 2002; 575–578.
15. Kwon S, Kim J, Kim C. An efficient tree structure for delay sensitive data aggregation in WSN, In *Proceedings of IEEE Advanced Information Networking And Applications*, 2008.
16. Tan H, Korpeoglu I, Stojmenovic I. Computing localized power efficient data aggregation trees for sensor networks. *IEEE Transactions on Parallel and Distributed Systems* 2011; **22**(3): 489–500.
17. Wu Y, Li X, Liu Y, Lou W. Energy-efficient wake-up scheduling for data collection and aggregation. *IEEE Transactions on Parallel and Distributed Systems* 2010; **21**(2): 275–287.
18. Yu Y, Krishnamachari B, Prasanna VK. Energy-latency tradeoffs for data gathering in wireless sensor networks, In *Proceedings of IEEE INFOCOM*, 2004.
19. Zhu Y, Sundaresan K, Sivakumar R. Practical limits on achievable energy improvements and usable delay tolerance in correlation aware data aggregation in wireless sensor networks, In *Proceedings of SECON*, 2005.
20. Xu X, Wang S, Mao X, Tang S, Li X. A delay efficient algorithm for data aggregation in multi-hop wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems* 2011; **22**(1): 163–175.
21. Bianchi G. Performance analysis of IEEE 802.11 distributed coordination function. *IEEE Journal of Selected Areas in Communications* 2000; **18**(3): 535–547.
22. Banchs A, Vollero L. A delay model for IEEE 802.11e EDCA. *IEEE Communications Letters* 2005; **9**(6): 508–510.
23. Liu J, Niu Z. Delay analysis of IEEE 802.11e EDCA under unsaturated conditions, In *Proceedings of IEEE WCNC*, 2007; 430–434.
24. Malone D, Duffy K, Leith D. Modeling the 802.11 distributed coordination function in non-saturated heterogeneous conditions. *ACM/IEEE Transactions on Network* 2007; **15**(1): 159–172.
25. Li K. Design and performance evaluation of communication algorithms in multihop wireless networks with multiple channels. *International Journal of Parallel, Emergent and Distributed Systems* 2010; **25**(6): 465–488.
26. Bajaj L, Takai M, Ahuja R, Tang K, Bagrodia R, Gerla M. GloMoSim: a scalable network simulation environment. *Technical Report 990027*, Computer Science Department, University of California, Los Angeles, 1999.


## AUTHORS' BIOGRAPHIES

**Xu Li** received a BSc (1998) degree from Jilin University, China, an MSc (2005) degree from the University of Ottawa, Canada, and a PhD (2008) degree from Carleton University, Canada, all in computer science. Before joining the National Institute for Research in Computer Science and Control (INRIA), France, as a full researcher, he held postdoctoral fellow positions at the University of Waterloo, Canada, INRIA/CNRS, France, and University of Ottawa, Canada. His research has been funded directly by the Natural Sciences and Engineering Research Council of Canada (NSERC). He has published more than 50 peer-reviewed papers. His current research interests are in the areas of machine-to-machine communications and Internet of Things. He is an editorial board member of the European Transactions on Telecommunications, Ad Hoc and Sensor Wireless Networks, and Parallel and Distributed Computing and Networks. He is a guest editor of *Computer Communications*, *Journal of Communications*, and *Peer-to-Peer Networking and Applications*. He is/was among the organizing committees and technical program committees of many major conferences, for example, ADHOC-NOW '08, 11, and 12; IEEE IPCCC '11, IEEE LCN '10 and 11; IEEE MASS '07 and 11, IEEE DCOSS '11, etc.

**Shuo Yan** is working toward this master's degree in the School of Information Technology and Engineering, University of Ottawa, Canada. His research interest is in QoS routing in wireless sensor networks.

**Chendong Xu** received his master's degree from the School of Information Technology and Engineering, University of Ottawa, Canada. His research interest is in data aggregation in wireless sensor networks.

**Amiya Nayak** received the BMath degree in computer science and combinatorics and optimization from the University of Waterloo in 1981 and the PhD degree in systems and computer engineering from Carleton University in 1991. He has more than 17 years of industrial experience, working at CMC Electronics, Defense Research Establishment Ottawa (DREO), EER Systems, and Nortel Networks, in software engineering, avionics and navigation systems, simulation, and system performance analysis. He is on the editorial board of the IEEE Transactions on Parallel and Distributed Systems, International Journal of Parallel, Emergent, and Distributed Systems, International Journal of Computers and Applications, International Journal of Computer Information Technology and Engineering, and International Journal of Computing and Information Science. He is the editor-in-chief of Parallel and Distributed Computing and Networks. He is currently a full professor in the School of Information Technology and Engineering, University of Ottawa. His research interests are in the areas of mobile ad hoc and sensor networks, fault tolerance, and distributed systems/algorithms, with more than 150 publications in refereed journals and conference proceedings.

**Ivan Stojmenovic** received the PhD degree in mathematics. He has held regular and visiting positions in Serbia, Japan, US, Canada, France, Mexico, Spain, United Kingdom (as a chair in applied computing at the University of Birmingham), Hong Kong, Brazil, Taiwan, and China, and is a full professor at the University of Ottawa, Canada, and an adjunct professor at the University of Novi Sad, Serbia. He has published more than 250 different papers, on wireless, ad hoc, sensor, and actuation networks, and has edited seven books on applied algorithms with Wiley. He is an editor of over dozen journals, editor-in-chief of *IEEE Transactions on Parallel and Distributed Systems* (since January 2010), and founder and editor-in-chief of three journals (*Journal of Multiple-Valued Logic and Soft Computing*, *International Journal of Parallel, Emergent and Distributed Systems*, and *Ad Hoc & Sensor Wireless Networks: An International Journal*). He is one of 260 computer science researchers with h-index $>$ 40 and has more than 10 000 citations. He received three best paper awards and the Fast Breaking Paper Award in October 2003 by Thomson ISI ESI. He is the recipient of the Royal Society Research Merit Award, United Kingdom. He received an Excellence in Research Award from the University of Ottawa in 2009. He chaired and/or organized more than 60 workshops and conferences and served on more than 200 program committees. He was a program co-chair at IEEE PIMRC 2008, IEEE AINA 2007, IEEE MASS 2004 and 2007, EUC 2005, 2008, 2009, and 2010, AdHoc-Now 2008, IFIP WSAN 2008, WONS 2005, MSN 2005 and 2006, and ISPA 2005 and 2007; founded workshop series at IEEE MASS, ICDCS, DCOSS, WoWMoM, ACM MobiHoc, IEEE/ACM CPSCom, FCST, and MSN; and is/was a workshop chair at IEEE INFOCOM 2011, IEEE MASS 2009, and ACM MobiHoc 2007 and 2008. He is a fellow of the IEEE (Communications Society, class of 2008) and is an IEEE Communications Society distinguished visitor from 2010–2012.