Deadline-Aware Data Collection in CSMA/CA-based Multi-Sink Wireless Sensor Networks

A. Hiromori†, A. Uchiyama†, H. Yamaguchi†and T. Higashino†

†Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
{hiromori, utiyama, h-yamagu, higashino}@ist.osaka-u.ac.jp

ABSTRACT

In this paper, we propose a new data collection method named DD (Drainage Divide), targeting wireless sensor networks having multiple sinks and sensors that periodically generate data. DD builds a set of disjoint spanning trees rooted at the sinks for data collection, where the delay from each node to the corresponding sink is kept to meet a given deadline. DD constructs a set of initial trees in a greedy fashion, and adjusts the trees in a centralized manner so as to meet the deadline by estimating the node-to-sink delay. For the forwarding delay estimation at each node, we have applied the CSMA/CA-based MAC delay analysis considering interference nodes. The simulation results have shown that DD avoids large delays by choosing further sinks instead of the congested nearest sinks.

Keywords: Wireless Sensor Networks, Data Collection, Multiple Sinks

1 INTRODUCTION

Some applications in wireless sensor networks (WSNs) aim to reliably collect real-time sensing data such as pulses, respiratory curves, pictures and videos [1], [2]. In such situation, deployment of multiple sinks could be assumed for traffic load balancing. While many QoS routing protocols in WSNs have been developed to cope with limited capacity, instability of network topology and channel contention, real-time data collection needs to be considered from the different aspects such as energy-efficiency, load-balancing, transmission scheduling and so on. Some methods utilize scheduling based on Time Division Multiple Access (TDMA) to avoid collision in packet transmission [3] and some others consider dynamic detouring to avoid congested regions [4], [5]. However, it is desirable to properly build delivery trees considering the impact of a collected data amount on forwarding delays for timely collection of periodically generated data that may be large and hence may cause considerable forwarding delays.

In this paper, we propose a novel routing protocol named DD (Drainage Divide) [1], targeting WSNs having multiple sinks and sensor nodes that periodically (at a short interval) generate data to be delivered to those sinks. Then DD builds a set of disjoint spanning trees rooted at the sinks for the data delivery, where the delay from each node to the corresponding sink is kept under a given threshold (i.e. deadline). For the construction of feasible trees under the delay constraint, we have applied the CSMA/CA-based MAC delay analysis [6] for estimating a forwarding delay at each node by the information about the current trees. DD constructs a set of initial trees in a greedy fashion, and adjusts the trees so as to satisfy the delay constraints. This is done by estimating the node-to-sink delay precisely based on the delay analysis and reconstructing the current trees in a centralized manner.

In order to evaluate the performance of DD, we have implemented DD in the QualNet simulator [7] and compared the performance of DD with that of a simple protocol which builds shortest path trees using AODV [8]. The results have shown that DD avoids large delays by choosing further sinks instead of the congested nearest sinks, to meet the deadline.

[1] This research was supported by JST, CREST.

2 RELATED WORK

2.1 QoS Routing in Mobile Wireless Ad-hoc Networks

Mobile wireless ad-hoc networks are self-organized networks constructed by mobile nodes. There are a number of papers that deal with QoS routing protocols in these networks in order to establish a route that satisfies end-to-end QoS requirement. For example, Ref. [9] analyzes the performance of some routing protocols and shows the effect of inaccurate information. It also shows efficient routing schemes that work well with inaccurate information. Sivakumar et al have proposed a distributed QoS routing protocol which can react quickly and effectively to network dynamics by maintaining and informing link status in ad-hoc networks [10]. A routing protocol considering multiple paths has also been proposed to improve the overall network performance [4]. Ref. [3] presents a novel routing protocol that utilizes a TDMA-based scheduling technique to satisfy an end-to-end delay or bandwidth requirement. Since these protocols are designed for unicast communication, it is not efficient to apply directly to the case that we have to deal with in this paper.

2.2 Data Collection in WSNs

There are many researches dealing with data collection in WSNs. Some techniques aim to collect sensing data with minimum delay so that applications can handle the “fresh” data like real-time medical monitor and surveillance.

One known socialized network topology management is PEGASIS [11]. PEGASIS constructs a chain topology not only to gather data efficiently but also to save energy consumption. The chain is divided into a number of groups to space out simultaneous transmissions. It means that a node communicates with at most two specified nodes and this enables to reduce wireless communication overhead drastically. As data-centric routing, Krishnamachari has summarized data gathering problems theoretically and has shown a heuristics algorithm with polynomial complexity [12]. SPEED also achieves real-time communication by considering geographical positions of nodes [13].

There are several researches that assume multiple paths or multiple sinks. A QoS provisioning technique is proposed that constructs and maintains several paths between two nodes based on local link state information [5]. Another paper assumes that there are several sink nodes on wireless sensor networks and formalizes the problem where nodes are clustered with a sink node [14]. It has also proposed distributed routing protocols to build clusters to reduce energy consumption.

On the other hand, there are other researches to handle sensing data with different data rates. RPAR [15] has achieved application-specified communication delays at low energy cost by dynamically adapting transmission power and routing decisions. Karenos et al have proposed the method that satisfies a packet delay constraint by adjusting transmission power and changing routes [16].

There are several techniques designed by combining some of the above techniques. Gao proposes a topology management method for sparsely deployed wireless sensor nodes [17]. To deal with the case, the method constructs nearly optimal network structure by selecting “hot nodes”, which are a part of delivery tree in order for communicating nodes to be able to find each other at the same time. In terms of multiple paths and QoS provisioning, by extending the original SPEED presented in Ref. [13], Felemban et al have proposed MMSPEED that constructs several layers so that it can handle several traffic classes [18].

2.3 Our Contribution

In our electronic triage project [1], we are designing and developing WSNs constituted of wireless nodes being capable of sensing vital signs. It is supposed that those vital signs are sent at different rates since those of heavily injured patients should be sampled frequently. Moreover, the size of the collected data may be large, which results in a forwarding delay at each node if we employ standard CSMA/CA-based MAC devices. Therefore, a method for real-time, periodic and continuous data collection is required.

Our contribution is to design a new protocol named DD that guarantees timely delivery of data in such a complex situation. DD satisfies a given delay constraint in such a case that forwarding delay is mainly caused by collection of data via trees. For this purpose, DD estimates forwarding delays at every hop considering interference caused by nearby nodes. We have applied a forwarding delay analysis by considering CSMA/CA channel contentions including interference nodes proposed in Refs. [6], [19].

3 PRELIMINARIES

3.1 Assumptions on Sensor Networks and Data Collection

3.1.1 Network Model

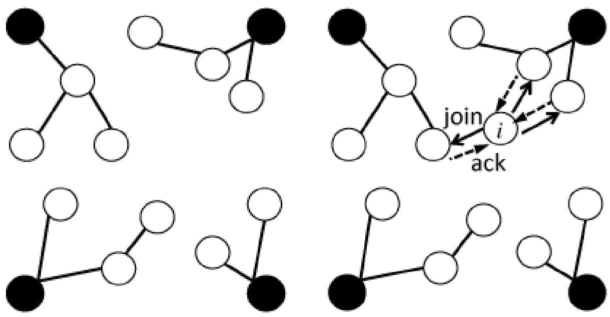
The networks targeted in this paper are wireless sensor networks (WSNs), each of which has multiple sinks called *base stations*. We assume that base stations are connected via wired/wireless links with each other. This means that base stations can share information that is collected from the network without additional network cost. Wireless sensor nodes, simply called *nodes*, have wireless devices that can communicate with neighbor nodes within a certain wireless radio range. Nodes may be mobile, while base stations are stationary. When a node transmits a piece of data, it takes a certain delay according to not only the data from the node but also those from its interference nodes. The details of the delay model will be given in Section 4.2.

3.1.2 Data Sensing and Collection

We assume that each node periodically monitors environments (e.g. vital signs of human bodies) and the obtained data is delivered to one of the base stations. We assign exactly one base station to each node at a time, but it may change over time by the proposed data collection algorithm. The data collection is done by a set of trees each of which is rooted at one of the base stations. The intermediate and leaf nodes are sensor nodes, and the trees are disjoint. Each node sends its own data to the base station by forwarding it to its unique parent node on the tree. This forwarding is done by unicast in the datalink layer.

3.2 Overview of DD Protocol

Our goal is to construct a set of data collection trees in which the obtained data are delivered to the base stations within a given delay bound, called *deadline*. We formalize the problem to construct a set of trees that



(a) Example Topology (b) Node i tries to join

Figure 1: Overview of DD Protocol

satisfy the delay constraint and propose a heuristic algorithm to this problem. We show how the proposed algorithm works briefly in Fig. 1. In the figure, there are four base stations denoted by black circles, and the others denoted by white circles are nodes. In the proposed protocol, a new node tries to connect itself to one of the existing trees in order to join the system (initially each tree includes one base station and no nodes). For example, Fig. 1 (b) exemplifies how a new node i joins an existing tree. At first, node i finds neighbor nodes by broadcasting a join message and the neighboring nodes reply acknowledgments that allow the node to connect to the trees. Then, node i chooses the one with the minimum delivery delay to the base station as its parent node and joins the tree.

In order to assess that the delay of data delivery from each node to the corresponding base station is below the deadline, DD protocol estimates the delay based on CSMA/CA-based MAC delay analysis [6]. It requires the topologies of the trees and the size of the data generated by each node since the protocol estimates delays caused by data transmissions based on these information. Therefore, after joining the tree, the node periodically sends its neighbor nodes, the amount of data generated by it per unit time and its current tree to the base station.

If a node does not satisfy the deadline, the tree is partially reconstructed. At first, in order to meet the deadline, the protocol tries to connect the node to a different tree so that the node can send its data to the base stations with smaller delay. For example, we assume that the node i can not send its data within the deadline in Fig. 1 (b). The protocol tries to find a new parent for node i that can accommodate node i and connect node i to the parent node. If the protocol can not find such parents, the protocol tries to reduce the traffic load of node i's parent node by moving node i's sibling nodes or the parent node itself to another tree since the delay is also affected by the amount of the network traffic generated by interference nodes and might be reduced by reducing the network traffic.

4 DEADLINE-AWARE FOREST CONSTRUCTION PROBLEM

4.1 Problem Formulation

Let us denote sets of sensor nodes and base stations by $N = \{n_1, n_2, ..., n_m\}$ and $B = \{b_1, b_2, ..., b_r\}$, respectively. We also denote a set of sensor nodes and base stations as $V = v_1, ..., v_{m+r}$. Therefore, $N \cap B = \emptyset$ and $N \cup B = V$. We represent the wireless range of nodes by R where we assume the range is common to all the nodes in V. We utilize the unit-disk model as the communication model where two nodes are communicable if the distance is not longer than R. The sensor network can be represented as $G = (V, E)$ where $E \subseteq V \times V$ is a set of potential edges. An edge $e_{ij} = (v_i, v_j)$ is in E if and only if nodes v_i and v_j are communicable. At a given, common interval, each node generates a piece of data, which should be delivered to one of the base stations within a given deadline denoted by DL. For simplicity, we assume that the size of the data from a node is the same over time.

Then we consider constructing a set of trees on G, which are disjoint with each other and rooted at the base stations. Such a set of trees on G is called a *forest*. Formally, for the forest $F = \bigcup_i T_i = (V_i, E_i)$, $\bigcup_i V_i = V$ and $V_i \cap V_j = \emptyset$. In each tree T_i, a piece of data generated by node v_i is forwarded via the datalink unicast to its parent node v_p along the tree. Therefore, the total amount of data $D(v_p)$ that node v_p should forward to its parent is given by;

$$D(v_p) = \sum_{v_k \in child(v_p)} D(v_k) + d(v_p) \quad (1)$$

where $d(v_p)$ denotes the data size generated by node v_p.

Hereafter, we let "$delay(e)$" denote a data forwarding delay function on a link e. This should be modeled considering the channel acquisition ratio and the bandwidth. The channel acquisition ratio is determined by the amount of data transmitted by interference nodes. Later we show a modeling of the delay characteristics according to the data size. The delivery delay of the data from node v to base station b can be given as follows;

$$SendTime(v) = \sum_{e \in path(v,b)} delay(e) \quad (2)$$

where $path(v, b)$ denotes the set of edges from node v to base station b on the tree. $path(v, b)$ can be calculated from the current forest F on G.

At last, we can define the decision problem of determining whether there exists a forest that satisfies the deadline constraint or not for a given network topology G and deadline DL. This problem is called *deadline-aware forest construction problem*. It is not easy to solve the problem straightforward because this problem contains to construct several spanning trees on a graph based on a performance metric. In addition, $deley(e)$ varies according to the tree topologies and the amount of the transmitted data on the trees. Therefore, we propose a heuristic algorithm for this problem in Section 5.

4.2 Delay Modelling based on CSMA/CA

Ref.[6] shows not only throughput analysis but also delay analysis under CSMA/CA. In this paper, we apply this analysis to derive the forwarding delay on link e. According to Ref.[6], the average transmission delay $delay(e)$ based on CSMA/CA can be obtained by the following equation.

$$delay(e) = E[N_c](E[BD] + T_C + T_O) + (E[BD] + T_S) \quad (3)$$

We denote the expected number of collisions until a transmission succeeds and the average backoff delay as $E[N_C]$ and $E[BD]$, respectively. We also denote the time for checking channel availability after a transmission collides, the average time to detect a collision and the average time to detect a packet transmission without collisions as T_O, T_C and T_S, respectively. These variables depend on the channel access method (i.e. with or without RTS/CTS). $E[BD]+T_C+T_O$ in the equation means the duration that a node sends its data and knows the transmission failed. The node repeats to send its data $E[N_c]$ times and it requires $E[N_c](E[BD] + T_C + T_O)$. After $E[N_c]$ times transmission, the node is expected to send its data successfully with $E[BD]+T_S$. $E[N_C]$ and $E[BD]$ can be determined based on the traffic amount of its interference nodes. Thus, we can estimate the transmission delays on each link precisely from the traffic amount of interference nodes for each node. This estimation contributes to construct trees that can meet deadline constraints without many probe packets that might cause network congestion.

5 DD PROTOCOL

Data collection using simple shortest paths to base stations may cause a large delay since packets are concentrated at some intermediate nodes and base stations and they may experience traffic congestion. In such a case, it is efficient to use a path to another base station, which may not be the closest base station, in order for traffic load balancing.

We propose a heuristic algorithm named DD (Drainage Divide) to construct trees for realtime data collection without aggregation in wireless sensor networks. DD protocol firstly constructs initial trees in the initial phase, and then modifies the constructed initial trees in reconstruction phases if a deadline violation occurs. The tree modification is computed at a server because topology information of whole networks is required to calculate delays from nodes to base stations. Delays at each link can be obtained by the function as described in the equation (3). This model needs the interference nodes and thier traffic amounts for each node to calculate its delay. We apply 2-hop nodes from a node on graph G as its interference nodes since it is difficult for a node to know its interference nodes precisely. From the equation (2), the end-to-end delay from node v to base station b can be also calculated as the total of delays at each link in the path between node v and base station b. For this calculation, each node periodically collects and sends information including its node identification, a list of neighbor nodes, the amount of data to be generated per unit time and its belonging tree information. The tree information of node v consists of an identification of base station b and the number of hops to base station b if node v belongs to the tree rooted at base station b. The tree information is null if node v does not belong to any tree. We will explain how DD protocol constructs trees on graph G in detail.

5.1 Initial Tree Construction

Initial trees are constructed to gather such information from all nodes so that the server can estimate the delay at each node. Node v broadcasts a hello packet to its neighbors at first. The neighbor nodes of node v reply their identification and tree information if they have already joined any tree. Thus, node v can build a list of its neighbor nodes based on the replies from the neighbor nodes. After that, if node v has received several valid tree information, node v selects a neighbor node for its parent node p_v where the number of hops to p_v's corresponding base station is the smallest. On the other hand, node v randomly selects its parent from neighbor nodes if there is no valid tree information.

Node v sends a list of its neighbor nodes to node p_v after the above operation. Node p_v forwards the information from node v to the parent of node p_v. In this step, a loop can be detected if node v receives the same information that is initially transmitted from node v. In that case, node v changes its parent node from node p_v to another neighbor node in the same way.

5.2 Tree Reconstruction Phase

When the server detects that a node in a tree can not satisfy the deadline, the tree is partially reconstructed. The tree reconstruction phase is processed as follows. For all leaf node l, the server calculates its $SendTime(l)$ periodically since once leaf nodes can transmit within the deadline, all other nodes are guaranteed to be able to transmit within the same deadline. In addition, the leaf nodes are located near other trees and might change their trees easily to meet the deadline. DD protocol starts to reconstruct the tree if there are some leaf nodes whose delays are more than DL. We describe a set of such nodes as N_v. DD protocol checks whether N_v can connect the other trees or reduce the traffic load of their parent nodes in the following ways.

Connecting to Other Neighbor Node Let us assume that leaf node l can not send its data within DL via its parent node p. At first, DD protocol finds a node p' from neighbor nodes of leaf node l such that $SendTime(p')$ is the smallest. In addition, each node v on $path(p', bs)$ has to meet its deadline so that node p' can be chosen for the parent node for leaf node l. If so, leaf node l can be removed from N_v because leaf node l may send its data within the deadline via new parent node p'. After this operation, node p may become a leaf node on the tree since node p may not have a child node anymore. Therefore, node p is added to N_v if node p is a leaf node, $SendTime(p)$ is more than the deadline and node p is also need to change its parent node. For all leaf nodes l in N_v, the above operations are executed. Despite of these operations, if there are still some nodes in N_v, DD protocol starts reducing the traffic load on the path from the node to the base station so that the delay on the path can be reduced.

Reducing Traffic Load on a Path We denote a common parent node of some nodes in N_v and the parent of the common parent node as node i and node p_i, respectively. We also represent $C_{i,j}$ and $Ne_{i,j}$ as the children of node i and the other neighbor nodes of node i, respectively (see Fig. 2(a)). From the definition, some nodes in $C_{i,j}$ belong to N_v.

In order to eliminate their deadline violations, the traffic load on the path from the nodes to their base station should be reduced. Therefore, DD tries to move some children nodes of node i to the other tree as shown in Fig. 2(b), and then node i connects another parent node (see Fig. 2(c)). As shown in the figure, the separated children nodes have to find a new parent node. For the separated children nodes, DD selects a new parent node from their neighbor nodes and connects the nodes to it so that their deadline constraints are not violated. This operation can reduce the traffic load of node i and its neighbor nodes and the delay on the path from the nodes to their base stations can be also reduced. DD repeats the above procedure in the reconstruction phase for the parent node p_i of node i until there is no nodes in N_v.

6 PERFORMANCE EVALUATION

6.1 Simulation Settings

We implemented DD on the QualNet [7] for simulation experiments. The simulation settings are summarized in Table 1. We compared delays of DD with those of AODV [8] in $1500m \times 1500m$ region with 50 nodes and 4 base stations. In Fig. 3, 4 stars represent the base stations. Since AODV is a unicast routing protocol, we assigned the nearest base station as a destination node to each sensor node. We also assume that sensor nodes send data via IEEE802.11a with 6.0Mbps transmission rate. Each sensor sends packets every 0.5 seconds, and we prepared 3 types of traffic classes: "low traffic class", "normal traffic class" and "heavy traffic class". Low, normal and heavy traffic class nodes send 1024 byte packets, 2048 byte packets and 3072 byte packets, respectively. In the following simulation scenarios, we have allocated one of these traffic classes to the set of nodes in each region.

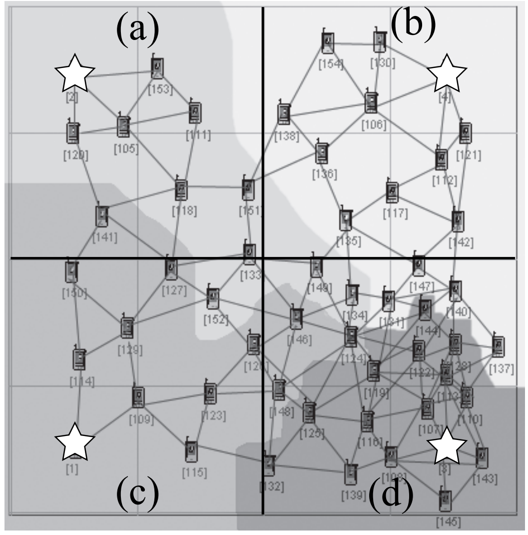


Figure 3: Node Deployment and DD Spanning Trees

Simulation Time	5 min
Area Size	1500m × 1500m
# of Base Station (BS)	4 nodes
BS Location (x,y)	(200,200), (200,1300), (1300,200), (1300,1300)
# of Sensor Node	50 nodes
Packet Payload Size	1024byte, 2048byte or 3072byte
Interval	0.5s
Deadline Time DL	0.1 or 0.15 sec
Radio Type	802.11a
Transmission Rate	6Mbps
Radio Range	270m

Table 1: Simulation Settings

6.2 Delay Comparison

Firstly, sensor nodes were deployed uniformly in the simulation area and we allocated "normal traffic class" to all nodes. The deadline was set to 0.1 sec. Fig. 4 shows average end-to-end delays for each node. We can see that long delays for some nodes in AODV are remarkably reduced in DD to meet the deadline for all nodes by shifting delays from other long delay nodes to nodes with shorter delays.

Secondly, nodes were nonuniformly located as shown in Fig. 3. The nodes in region (d) were densely deployed compared with other regions. The nodes located in regions (a) and (d) were allocated to "low traffic class"

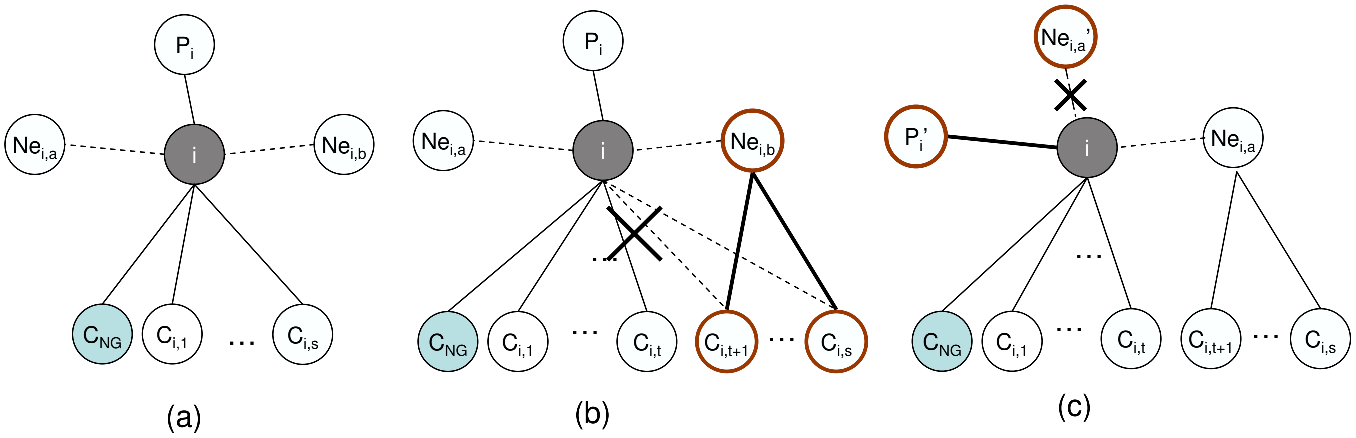


Figure 2: Reduction the Traffic Load of parent; (a) Original Tree; (b) Move Children Node of Node i; (c) Switch Parent Node of Node i

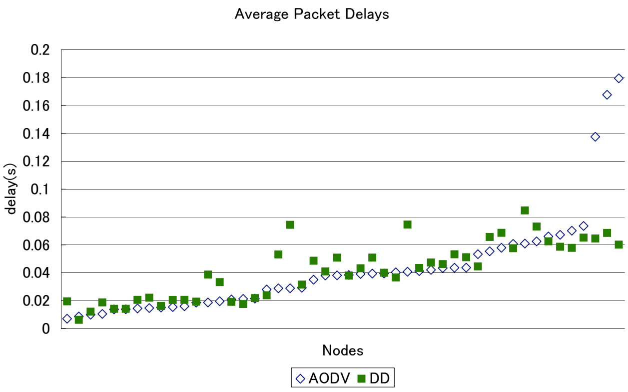


Figure 4: Packet Delays in Uniform Deployment

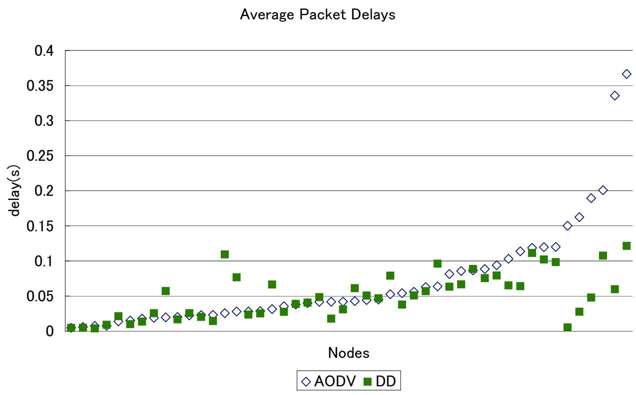


Figure 5: Packet Delays in Nonuniform Deployment

and "heavy traffic class", respectively. Other nodes were allocated to "normal traffic class". Here, the deadline was set to 0.15 sec. Fig. 3 shows the areas covered by each spanning tree, and Fig. 5 shows average end-to-end delays for each node. 12% of nodes in AODV exceed the deadline due to congestion caused by dense deployment and heterogeneous traffic patterns compared to the uniform deployment. On the other hand, no node in DD violates the deadline. From the result, we can see that DD can handle the situation where nodes send different data rates, and nodes are deployed heterogeneously.

From above two different simulation scenarios, we have confirmed DD can construct a set of spanning trees considering node deployment and different traffic trends for each case.

7 CONCLUSION

In this paper, we have proposed a new routing protocol named DD (Drainage Divide) for wireless sensor networks. DD is targeting wireless sensor networks having multiple sinks and sensor nodes that periodically (at a short interval) generate data to be delivered to those sinks. DD builds a set of disjoint spanning trees rooted at the sink nodes for the data delivery, where the delay from each node to the corresponding sink node is kept under a given threshold called deadline. We have applied the CSMA/CA-based MAC delay analysis so that we can estimate forwarding delays at each node by the information about the current trees. DD constructs a set of initial trees in a greedy fashion, and adjusts the trees so as to satisfy the delay constraints. This is done by estimating the node-to-sink delay precisely based on the delay analysis and reconstructing the current trees in a centralized manner without many probe packets that might cause network congestion. The simulation results have shown that the delay for data collection in DD satisfies 100% of deadline constraints while the compared AODV-based routing violates 12% of the constraints. As future work, we are designing a distributed protocol considering multiple routes to avoid congestion.

REFERENCES

[1] N. Kiyama, J. Kusuda, A. Uchiyama, A. Hiromori, T. Umedu, H. Yamaguchi, and T. Higashino. An advanced electronic triage system for rescue operations in disasters. In *Proc. of IPSJ Multimedia, Distributed, Cooperative, and Mobile Symposium*, pages 1837–1848, 2009.

[2] Stanislava Soro and Wendi Heinzelman. A survey of visual sensor networks. *Advances in Multimedia*, 2009, 2009.

[3] Chenxi Zhu and M. Scott Corson. QoS routing for mobile ad hoc networks. In *Proc. of 21st Annual Joint Conference of the IEEE Computer and Communications Societies(IEEE INFOCOM 2002)*, volume 2, pages 958–967, 2002.

[4] Shigang Chen and Klara Nahrstedt. Distributed quality-of-service routing in ad hoc networks. *IEEE Journal on Selected areas in Communications*, 17(8):1488–1505, 1999.

[5] Xiaoxia Huang and Yuguang Fang. Multiconstrained QoS multipath routing in wireless sensor networks. *Wireless Networks*, 14(4):465–478, 2008.

[6] Eustathia Ziouva and Theodore Antonakopoulos. CSMA/CA performance under high traffic conditions: throughput and delay analysis. *Computer Communications*, 25(3):313–321, February 2002.

[7] Qualnet. http://www.scalable-networks.com/.

[8] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc on-demand distance vector routing. In *Proc. of 2nd IEEE Workshop on Mobile Computing Systems and Applications(WMCSA '99)*, pages 90–100, 1999.

[9] Roch Guerin and Ariel Orda. QoS based routing in networks with inaccurate information: theory and algorithms. In *Proc. of 16th Annual Joint Conference of the IEEE Computer and Communications Societies(IEEE INFOCOM '97)*, volume 1, pages 75–83, Apr 1997.

[10] Raghupathy Sivakumar, Prasun Sinha, and Vaduvur Bharghavan. CEDAR: A core-extraction distributed ad hoc routing algorithm. *IEEE Journal on Selected Areas in Communications*, 17(8):1–12, 1999.

[11] Stephanie Lindsey and Cauligi S. Raghavendra. PEGASIS: Power-efficient gathering in sensor information systems. In *Proc. of IEEE Aerospace Conference*, volume 3, 2002.

[12] Bhaskar Krishnamachari, Deborah Estrin, and Stephen B. Wicker. The impact of data aggregation in wireless sensor networks. In *Proc. of 22nd International Conference on Distributed Computing Systems(ICDCSW '02)*, pages 575–578, 2002.

[13] Tian He, John A Stankovic, Chenyang Lu, and Tarek Abdelzaher. Speed: A stateless protocol for real-time communication in sensor networks. In *Proc. of 23rd International Conference on Distributed Computing Systems (ICDCS 2003)*, pages 46–55, 2003.

[14] Tommaso Melodia, Dario Pompili, Vehbi C. Gungor, and Ian F. Akyildiz. A distributed coordination framework for wireless sensor and actor networks. In *Proc. of 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing(MobiHoc '05)*, pages 99–110, 2005.

[15] Octav Chipara, Zhimin He, Guoliang Xing, Qin Chen, Xiaorui Wang, Chenyang Lu, John Stankovic, and Tarek Abdelzaher. Real-time power-aware routing in sensor networks. In *Proc. of 14th IEEE International Workshop on Quality of Service(IWQoS 2006)*, pages 83–92, 2006.

[16] Kyriakos Karenos and Vana Kalogeraki. Real-time traffic management in sensor networks. In *Proc. of 27th IEEE International Real-Time Systems Symposium(RTSS '06)*, pages 422–434, 2006.

[17] Jie Gao, Leonidas Guibas, Nikola Milosavljevic, and John Hershberger. Sparse data aggregation in sensor networks. In *Proc. of 6th Innternational Conference on Information Processing in Sensor Networks(IPSN '07)*, pages 430–439, 2007.

[18] Emad Felemban, Student Member, Chang gun Lee, and Eylem Ekici. MMSPEED: Multipath multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 5:738–754, 2006.

[19] Sumit Rangwala, Ramakrishna Gummadi, Ramesh Govindan, and Konstantinos Psounis. Interference-aware fair rate control in wireless sensor networks. *SIGCOMM Computing Communication Review*, 36(4):63–74, 2006.