# Dynamic Water Gate Assignment Scheme for Data Aggregation in Long-Thin Sensor Networks

You-Chiun Wang, Che-Hsi Chuang, Yu-Chee Tseng
*Department of Computer Science*
*National Chiao-Tung University*
*Hsinchu 30010, Taiwan*
*E-mail: {wangyc, chuangch, yctseng}@cs.nctu.edu.tw*

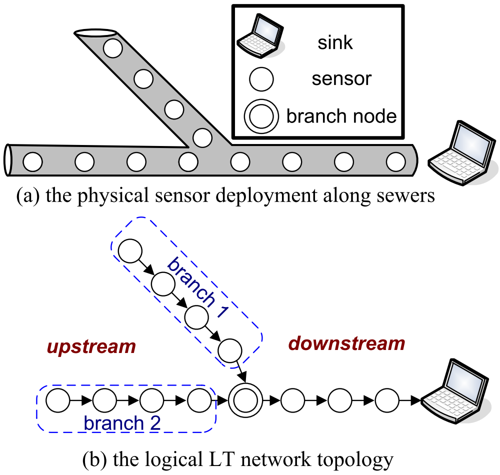
Chien-Chung Shen
*Department of Computer and Information Sciences*
*University of Delaware*
*Newark, DE 19716, USA*
*E-mail: cshen@cis.udel.edu*

***Abstract*—The *long-thin (LT)* network topology is promoted for many deployments of *wireless sensor networks (WSNs)* due to the application requirements. An LT topology consists of long branches of sensors, where each sensor has only one potential parent toward the sink. Data aggregation may help reduce excessive packet contention, but constraints imposed by the maximum payload size of each packet severely limit the amount of sensing data that can be aggregated along a long branch of sensors. Therefore, the paper suggests that multiple aggregation nodes, called *water gates*, should be designated along a branch to aggregate sensing data sent from their upstream sensors. Then, we develop a dynamic water gate assignment scheme that reduces the response time while avoids network congestion for data collection in LT WSNs, which can accommodate the time-varying sensing data generating rates of sensors. A testbed of 100 sensors is deployed and experimental results demonstrate the effectiveness of our scheme.**

***Keywords*-data aggregation; long-thin network; pervasive computing; wireless sensor network.**

## I. Introduction

*Wireless sensor networks (WSNs)* facilitate pervasive monitoring of the physical environments to enable many military and civil applications [1]–[4]. Sensors may be deployed to form networks with arbitrary topologies [5], [6] or regular topologies [7], [8]. In recent research, the *long-thin (LT)* network topology has been promoted for many WSN applications where the sensor deployment is subject to application requirements [9]. Practical examples include a $CO_2$ monitoring system inside tunnels, a surveillance system of moving cars along streets, and a monitoring system of water quality within sewers. Fig. 1(a) gives the physical sensor deployment along sewers, and Fig. 1(b) shows its logical LT network topology. Specifically, an LT topology consists of a bunch of long *branches* and each branch may have tens or even hundreds of sensors. For each sensor along a branch, there is only one potential parent node toward the sink. Branches are connected at *branch nodes*, which are denoted by double circles in Fig. 1(b).



(a) the physical sensor deployment along sewers

(b) the logical LT network topology

Figure 1: An example of the LT WSN deployed along sewers.

By viewing an LT WSN with $n$ sensors as a shortest-path tree rooted at the sink, let $d_i$ denote the depth of sensor $s_i$ from the sink, $i = 1..n$. Considering that each sensor transmits one packet to the sink without data aggregation, the WSN requires $\sum_{i=1}^{n} d_i$ transmissions to collect all sensing data. Fig. 1(b) gives an example, where the WSN requires 62 transmissions without data aggregation. Then, let us consider the following aggregation scheme: The leaf sensors begin transmitting their sensing data first. Each intermediate sensor waits to aggregate its own sensing data with the (aggregated) data sent from its child(ren), and then forwards the aggregated packet to the sink. Assume that successively aggregated sensing data can be loaded into one large packet, this scheme requires only $n$ transmissions. Fig. 1(b) shows an example, where the WSN requires 12 transmissions using this aggregation scheme. Nevertheless, constrained by the maximum payload size $L_{\max}$ of each packet and the compression ratio $\delta$ ($0 < \delta \leq 1$), the above scheme is not feasible for LT WSNs because each sensor can only aggregate up to $\lfloor \frac{L_{\max}}{\delta} \rfloor$ bytes of sensing data, while the total size of sensing data generated by the sensors along a long branch always exceed this bound.

To comply with the maximum payload size, this paper suggests that along a branch, multiple aggregation nodes (called *water gates*) should be designated, where each water

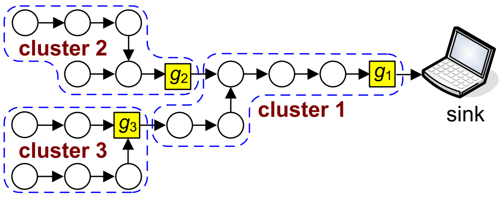


Figure 2: An LT WSN with three clusters.

gate aggregates its own sensing data with all of the sensing data from its upstream sensors (up to the immediate, upstream water gate(s) or the end of the branch) subject to the $\lfloor \frac{L_{\max}}{\delta} \rfloor$ bound. Fig. 2 shows an example. Considering that $L_{\max}$ is 3 bytes, $\delta$ is 0.5, and the size of sensing data is 1 byte, we designate one water gate every 6 sensors so that water gate $g_1$ aggregates its own sensing data with those from its 5 upstream nodes into one aggregated packet. Similarly, this can be done by water gates $g_2$ and $g_3$. Such an aggregated packet (of payload size $L_{\max}$) is said to be completely *filled up* with sensing data and is forwarded to the sink without being aggregated with any other sensing data along the branch. Besides, water gates help reduce contention among traffics by spatially separating areas where packets are transmitted. In Fig. 2, since water gates $g_2$ and $g_3$ hold their transmissions until enough sensing data from their upstream sensors have been collected, the sensors in cluster 1 can transmit their sensing data to $g_1$ with almost no interference from other sensors in clusters 2 and 3.

However, because sensors may generate sensing data at different *rates*, each water gate should wait for a different amount of time to completely fill up an aggregated packet of size $L_{\max}$. Let $\lambda_j$ denote the sensing data generating rate (or *data rate* for short) of sensor $s_j$ and $c(g_i)$ denote the *cluster* of nodes containing water gate $g_i$ and its upstream sensors. Assuming that each $\lambda_j$ is a constant and the WSN operates at a steady state, Eq. (1) indicates the amount of time $t_i$ taken by water gate $g_i$ to completely fill up an aggregated packet of size $L_{\max}$:

$$t_i = \frac{L_{\max}}{\delta \times \sum_{s_j \in c(g_i)} \lambda_j}. \tag{1}$$

When $t_i$ is large, the sink will wait for a longer time to receive an aggregated packet from $g_i$, which increases the network response time. Large $t_i$ implies either the size of $c(g_i)$ is small or the total data rate of the sensor nodes in $c(g_i)$ is low. On the contrary, small $t_i$ implies either the size of $c(g_i)$ is large or the total data rate of the sensors in $c(g_i)$ is high. In this case, too many packet transmissions will congest the network.

Therefore, this paper proposes a *dynamic Water gate Assignment scheme for Data Aggregation (WADA)* to facilitate effective data aggregation in LT WSNs. Given time thresholds $t_{lower}$ and $t_{upper}$ specified by the WSN application to reduce the network response time while avoid the congestion caused by transmitted packets, WADA designates water gates in an LT WSN such that for each water gate $g_i$, we have $t_{lower} \leq t_i \leq t_{upper}$. Moreover, when data rate $\lambda_i$ varies over time, WADA can adaptively adjust the locations of water gates to balance response time and congestion.

In the literature, the subject of data aggregation in WSNs has been extensively studied. However, most studies adopt either a tree-based or a cluster-based structure to aggregate data in WSNs, and none of them consider the LT topology. Tree-based aggregation schemes construct the shortest-path routing trees, and focus on how to choose a good routing metric to facilitate data aggregation. For example, the studies of [10], [11] propose data-centric schemes to select an appropriate path to reduce energy consumption. The study in [12] constructs an aggregation tree based on sensors' energy consumption. Each sensor predicts the energy consumption of its potential parents and selects the one that can remain the most energy as its parent. In contrast, there exists at most one route from a sensor to the sink in LT WSNs (*i.e.*, each sensor has at most one potential parent node toward the sink) so that existing solutions may not be directly applied.

In comparison, cluster-based aggregation schemes group sensors into clusters and conduct data aggregation within each cluster. For example, in LEACH [13] and PEGASIS [14], sensors relay the sensing data to their cluster heads, which are assumed to be able to directly communicate with the sink. However, this assumption is not valid in LT WSNs. HEED [15] groups sensors into clusters such that sensors within a cluster are single-hop away from the cluster head. However, in an LT WSN, sensors within a cluster are located along a long branch that are multi-hop away from their cluster head. SCT [16] proposes a ring-sector division clustering scheme, where sensors in the same section are grouped into one cluster. Explicitly, this scheme cannot be used in LT WSNs. To the best of our knowledge, the proposed WADA scheme is the first effort that addresses efficient data aggregation in LT WSNs.

We organize the rest of this paper as follows: Section II proposes our WADA scheme. Section III reports our prototyping experience and experimental results. Section IV concludes the paper.

## II. The Proposed WADA Scheme

An LT WSN can be modeled as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ includes the sink and all sensors, and $\mathcal{E}$ includes all of the communication links. We represent the topology of $\mathcal{G}$ as a tree rooted at the sink. Each sensor $s_j$ has a data rate of $\lambda_j$ that may vary over time, $j = 1..n$. A sensor is called a *branch node* if it has more than one child in $\mathcal{G}$. Fig. 2 shows an example, where there are three clusters and sensors $g_1$, $g_2$, and $g_3$ are water gates.

Given an LT WSN, WADA arbitrarily groups sensors into non-overlapping clusters and assigns their water gates. Note that the sensor nearest to the sink is always assigned as a water gate. On generating the sensing data, each regular sensor will send the data to its water gate. Each water gate $g_i$ then collects the sensing data from the sensors within its cluster $c(g_i)$. After collecting sufficient sensing data that can be aggregated to fill up one packet of payload size $L_{\max}$, $g_i$ sends the aggregated packet to the sink. To reduce the latency of waiting to aggregate sensing data to fill up one packet, $g_i$ dynamically adjusts its cluster based on the duration $t_i$ that it took to generate the last aggregated packet (by Eq. (1)). When $t_i < t_{lower}$, the total data rate within this cluster (*i.e.*, $\sum_{s_j \in c(g_i)} \lambda_j$) becomes too high, and the aggregated packets are sent to the sink more often. In this case, $g_i$ *shrinks* its cluster by excluding some sensors to reduce the total data rate. On the other hand, when $t_i > t_{upper}$, the total data rate within this cluster becomes too low, and the monitoring quality degrades. Therefore, $g_i$ *expands* its cluster by including more sensors to reduce the latency of generating aggregated packets. Note that within each cluster $c(g_i)$, the sensing data sent from each sensor $s_j \in c(g_i)$ may be relayed to $g_i$ in a pipeline manner subject to the contention of wireless transmissions. Thus, right before $g_i$ sends out each aggregated packet to the sink, the ratio of sensing data received from sensor $s_j$ within this packet is approximately equal to $\frac{\lambda_j}{\sum_{s_k \in c(g_i)} \lambda_k}$. That is, WADA ensures that the amount of reported sensing data from each sensor can be fairly proportional to the data rate of that sensor.

We then define the terms used in the paper. A water gate $g_k$ is called a *child water gate* of another water gate $g_i$ if $g_k$ is an immediate upstream water gate of $g_i$. In this case, $g_i$ is the *parent water gate* of $g_k$. Fig. 2 gives an example, where $g_2$ is a child water gate of $g_1$ and $g_1$ is the parent water gate of $g_2$. A water gate may have several child water gates but has only one parent water gate. Besides, if a water gate has no child water gate, it is called a *leaf water gate*; otherwise, it is called an *intermediate water gate*.

From an initial water gate designation, each water gate executes WADA asynchronously but will coordinates with its parent and child water gates. Specifically, each water gate $g_i$ measures its current $t_i$ value, moves one of its child water gates downstream or upstream by one hop if needed, and calculates its new $t_i$ value. This process is repeated until $g_i$ calculates that $t_{lower} \leq t_i \leq t_{upper}$. In particular, for each intermediate water gate $g_i$, two cases are considered:

- **Case of $t_i < t_{lower}$:** Water gate $g_i$ shrinks its cluster by querying each of its child water gates $g_k$ for its $t_k$ value. When $g_k$ is also adjusting its own child water gates, it replies to $g_i$ that itself is *busy*; otherwise, $g_k$ replies to $g_i$ for its $t_k$ value. When $g_i$ finds that all of its child water gates are busy, it waits for a $\Delta_t$ time and will try again. Otherwise, $g_i$ sends a *pull* message to the non-busy child water gate $g_k$ with the maximum $t_k$ value and whose parent node, say, $s_j$ is not a branch node. On receiving the pull message, $g_k$ asks $s_j$ to become a new water gate and clears itself as a water gate. Therefore, the old cluster $c(g_k)$ is replaced by a new cluster $c(s_j)$. Here, we use the term "move" to represent the above operation. However, when $g_i$ cannot find such a child water gate (which means that the parent nodes of all its non-busy child water gates are branch nodes), the non-busy child water gate $g_k$ with the maximum $t_k$ value is asked to move one-hop downstream. These operations are repeated until $g_i$ finds that $t_i \geq t_{lower}$. Fig. 3(a) shows an example, where $g_1$ wants to adjust its child water gates. Since $g_2$'s parent node is a branch node, $g_3$ is asked to move downstream. However, if the parent nodes of all child water gates are branch nodes, as shown in Fig. 3(b), $g_1$ asks $g_2$ to move to branch node $b$ (assuming $t_2 > t_3$).

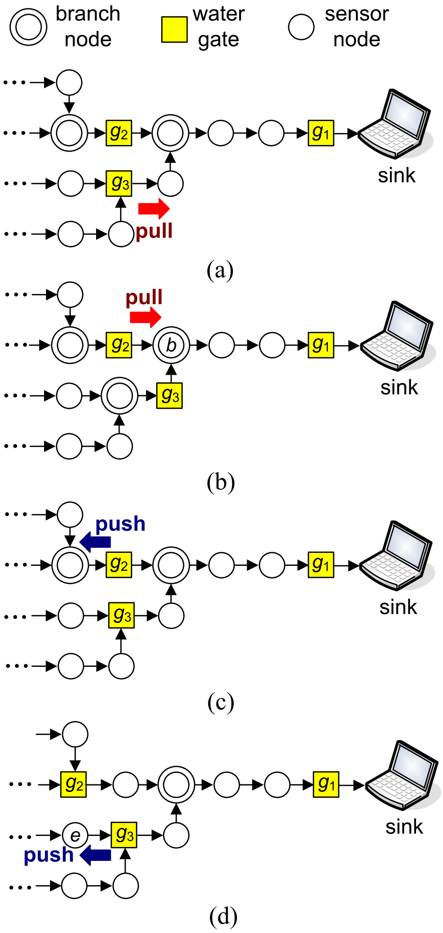
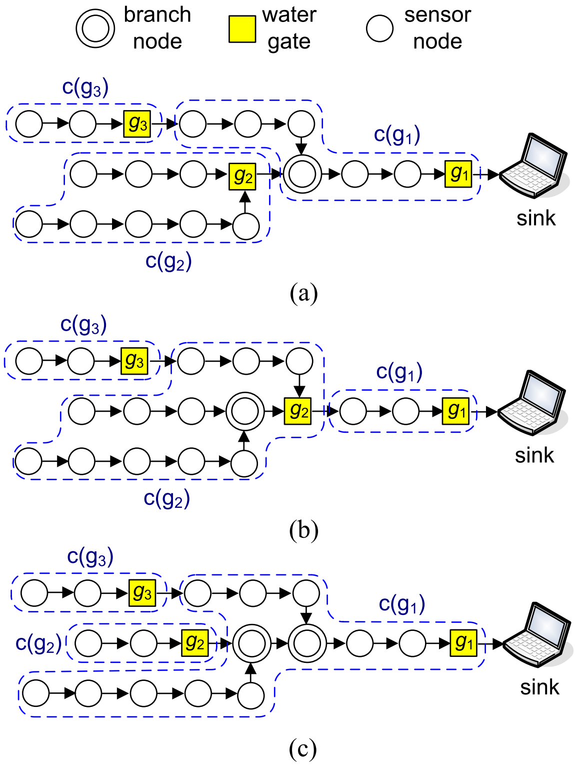


Figure 3: Four examples to show how to move child water gates: (a) $g_1$ pulls $g_3$ downstream due to $t_1 < t_{lower}$, (b) $g_1$ pulls $g_2$ to the branch node $b$ due to $t_1 < t_{lower}$, (c) $g_1$ pushes $g_2$ upstream due to $t_1 > t_{upper}$, and (d) $g_1$ pushes $g_3$ to node $s$ due to $t_1 > t_{upper}$.

To avoid excluding too many sensors when shrinking a cluster, which may make its $t_i$ value increase drastically, WADA avoids moving a child water gate whose



(a)

(b)

(c)

Figure 4: Bad examples of moving child water gates.

parent node is a branch node. Fig. 4(a) shows a bad example, where $t_2 > t_3$. If $g_1$ simply moves its child water gate with the maximum $t_i$ value, $g_2$ will be asked to move downstream, as shown in Fig. 4(b). In this case, the size of cluster $c(g_1)$ drastically decreases from 7 to 3. Moreover, the size of cluster $c(g_2)$ drastically increases from 9 to 13. Instead, WADA moves $g_3$ one-hop downstream.

- **Case of $t_i > t_{upper}$:** Water gate $g_i$ expands its cluster by querying each child water gate $g_k$ for its $t_k$ value. Then, $g_i$ sends a *push* message to ask the non-busy child water gate $g_k$ that is not a branch node and has the minimum $t_k$ value to move one-hop upstream. When $g_i$ cannot find such a child water gate (which means that all of its non-busy child water gates are branch nodes), the non-busy child water gate $g_k$ with the minimum $t_k$ value is asked to move upstream. These operations are repeated until $g_i$ calculates that $t_i \leq t_{upper}$. Fig. 3(c) gives an example, where $g_1$ wants to adjust its child water gates. Since $g_2$ is not a branch node, it will be moved upstream. When all child water gates are branch nodes, as shown in Fig. 3(d), WADA asks $g_3$ to move to node $e$ (assuming that $t_3 < t_2$).

To avoid including too many sensors when expanding a cluster, which may make its $t_i$ value decrease drastically, WADA avoids moving a child water gate that is a branch node. Fig. 4(a) shows a bad example, where $t_2 < t_3$. If $g_1$ simply moves its child water gate with the minimum $t_i$ value, $g_2$ will be asked to move upstream, as shown in Fig. 4(c). In this case, the size of cluster $c(g_1)$ drastically increases from 7 to 13. Moreover, the size of cluster $c(g_2)$ drastically decreases from 9 to 3. Instead, WADA moves $g_3$ one-hop upstream.

When moving a leaf water gate $g_i$, two cases should be considered:

- If $g_i$ is asked to move downstream but finds that $t_i < t_{lower}$, it selects one leaf node, say, $s_j$ in its cluster and designates $s_j$ as a new water gate. Thus, the number of water gates will increase by one.
- If $g_i$ is a leaf node but is still asked to move upstream, $g_i$ clears itself as a water gate. Thus, the number of water gates will decrease by one.

To avoid oscillating water gates, each water gate $g_i$ should maintain a list recording its past locations on $\mathcal{G}$. When $g_i$ finds that it has moved between two adjacent *oscillating nodes* more than $\alpha$ times and its parent water gate still asks it to move to one of the oscillating nodes, it enters the *oscillating state*. Then, $g_i$ notifies its parent water gate of stopping moving it in that direction. Water gate $g_i$ will exit the oscillating state if its parent water gate asks it to move to one non-oscillating node or a predefined timer expires.

## III. Experimental Results

We deploy 100 sensors to collect the environmental data, as shown in Fig. 5. Each sensor has a Jennic chip [17], which is a low power, wireless microcontroller supporting the ZigBee protocol [18]. Each sensor has a communication distance of 30 cm but two adjacent sensors are placed with a distance of 15 cm to ensure the network connectivity. We consider two LT topologies: $x$-topology and $y$-topology. As shown in Fig. 6, $x$-topology has four branches, where three branches have 22 nodes while one branch has 33 nodes; $y$-topology has three branches, where branch 1 has 15 node, branch 2 has 33 nodes, and branch 3 has 51 nodes. We compare WADA against the *Direct Relay (DR) scheme* and the *Static Water gate Assignment (SWA) scheme*. DR does not adopt any data aggregation and each sensor simply relays the sensing data from its upstream nodes to the sink. In SWA, each branch node is assigned as a water gate and we do not adjust the assignment of water gates during the experiments.

The packet size is 15 bytes, which contains a header of 12 bytes and a payload of 3 bytes. Each sensor reports its sensing data every $\Delta_s \in [7, 13]$ seconds and $\Delta_s$ will change every 30 seconds. We remove the headers of all received packets and concatenate their payloads into one single packet, so we have $\delta = 1$. In our experiments, we set $L_{\max}$ to 118 bytes, which is the maximum payload size defined in ZigBee. The experiment time is 10 minutes. In the experiments of executing WADA, the measurement

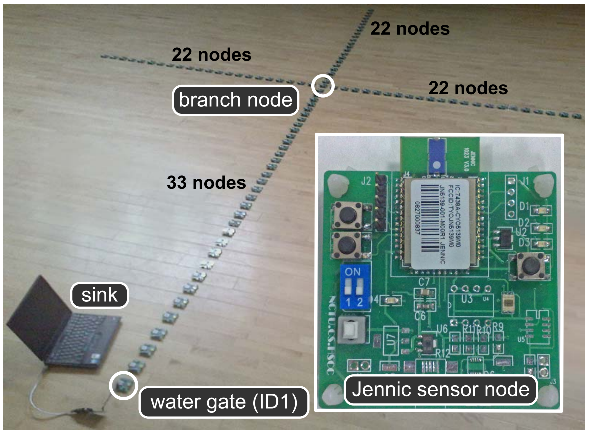
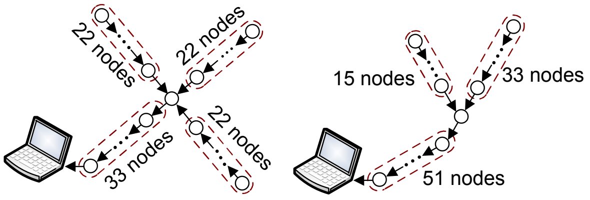
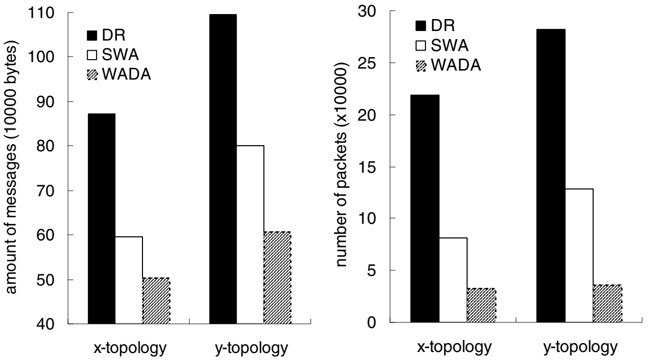


Figure 5: The snapshot of our 100-node LT WSN prototype.



(a) x-topology (b) y-topology

Figure 6: Two LT network topologies used in our experiments.



(a) amount of messages (b) number of packets

Figure 7: Comparison of data transmissions under $x$-topology and $y$-topology.

of messages sent by sensors includes all of the control messages such as query, reply, push, and pull used to adjust the assignment of water gates. Besides, we set $\Delta_t$ = 2 seconds, $t_{lower}$ = 24 seconds, and $t_{upper}$ = 26 seconds.

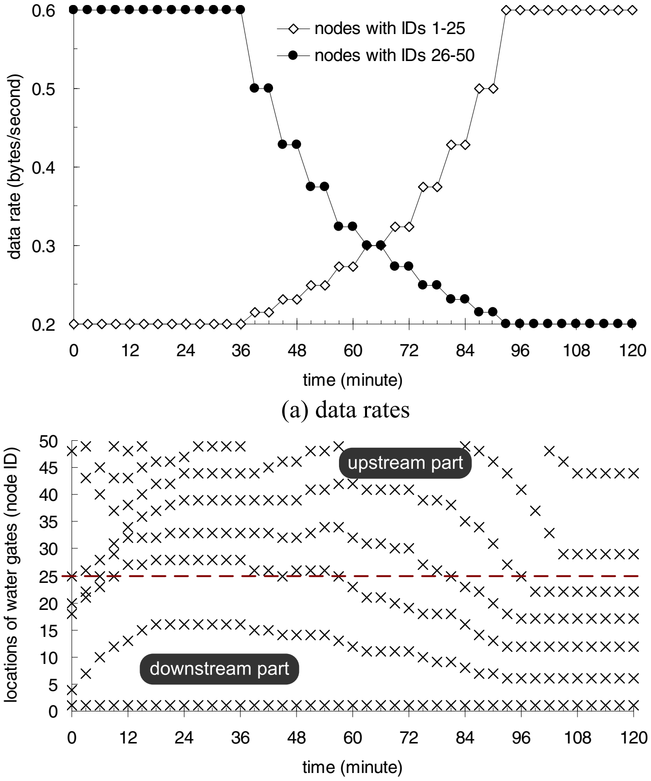
Fig. 7(a) shows the total amount of messages sent by sensors under different network topologies. All schemes incur higher amount of messages under the $y$-topology because this topology has the longest branch (with 51 nodes). DR incurs the highest amount of messages because it does not adopt any data aggregation. By dynamically adjusting the locations of water gates based on the network condition, WADA incurs the smallest amount of messages compared with DR and SWA. Specifically, under $x$-topology, WADA reduces 42.3% and 15.7% of message transmissions compared with DR and SWA, respectively. Under $y$-topology, WADA reduces 44.7% and 24.3% of message transmissions compared with DR and SWA, respectively. The above results verify the effectiveness of WADA.

Fig. 7(b) shows the total number of packets sent by sensors under different network topologies. When sensors transmit more packets, the network could be seriously congested. DR makes the sensors transmit the most number of packets, because it does not aggregate any sensing data. WADA incurs the smallest number of packets among all schemes because it adaptively clusters sensors and aggregates their packets accordingly. Under $x$-topology, WADA reduces 84.9% and 59.5% of packets compared with DR and SWA, respectively. Under $y$-topology, WADA reduces 87.3% and 72.2% of packets compared with DR and SWA, respectively. These results demonstrate that WADA significantly reduces the number of packets sent by sensors, which can greatly alleviate the network congestion.

To demonstrate the adaptability of WADA to varying data rates, we deploy a sink (of ID 0) and a line of 50 sensor nodes (of IDs 1 to 50), where the node with ID 1 is the most downstream sensor. The experiment time is 120 minutes and we measure the number of water gates and their locations over time. The $\lambda$ value of sensors with IDs 1 to 25 increases while that of sensors with IDs 26 to 50 decreases over time, as shown in Fig. 8(a). Here, we use the terms "downstream part" and "upstream part" to represent the sensors with IDs 1 to 25 and with IDs 26 to 50, respectively. Fig. 8(b) shows the locations of water gates over time. We observe that before the 60th minute, most water gates are located at the upstream part because sensors in the upstream part have a higher data rate. Therefore, WADA shrinks the sizes of clusters in the upstream part and thus assigns more water gates. After the two data rates cross around the 66th minute, the above behavior reverses. Most water gates move to the downstream part since sensors in the downstream part now have a higher data rate.

## IV. Conclusions

A large number of WSN applications dictate the deployment of the LT topology that demands new data aggregation solutions. This paper proposes the WADA scheme that assigns multiple water gates to regulate data aggregation and adapts the designation of water gates dynamically in response to changing data rates of sensors. WADA not only reduces the network response time but also avoids potential network congestion for data collection in LT WSNs. By using the sensors nodes equipped with Jennic wireless microcontrollers, we measure the performance of WADA through several experiments of our prototyped LT WSNs. Experimental results verify that WADA significantly reduces



(a) data rates

(b) the locations of water gates

Figure 8: The locations of water gates (marked by ‘x’) under different data rates.

the amount of data transmissions of sensors and adaptively adjusts the assignment of water gates according to various network conditions.

## Acknowledgement

Y.-C. Tseng’s research is co-sponsored by MoE ATU Plan, by NSC grants 97-3114-E-009-001, 97-2221-E-009-142-MY3, 98-2219-E-009-019, and 98-2219-E-009-005, by MOEA 98-EC-17-A-02-S2-0048, and 98-EC-17-A-19-S2-0052, by ITRI, Taiwan, by III, Taiwan, and by Intel.

## References

[1] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen, “The gator tech smart house: a programmable pervasive space,” *IEEE Computer*, vol. 38, no. 3, pp. 50–60, 2005.

[2] Y. C. Tseng, Y. C. Wang, K. Y. Cheng, and Y. Y. Hsieh, “iMouse: an integrated mobile surveillance and wireless sensor system,” *IEEE Computer*, vol. 40, no. 6, pp. 60–66, 2007.

[3] L. W. Yeh, Y. C. Wang, and Y. C. Tseng, “iPower: an energy conservation system for intelligent buildings by wireless sensor networks,” *International Journal of Sensor Networks*, vol. 5, no. 1, pp. 1–10, 2009.

[4] S. C. Hu, Y. C. Wang, C. Y. Huang, and Y. C. Tseng, “A vehicular wireless sensor network for co$_2$ monitoring,” in *IEEE Conference on Sensors*, 2009, pp. 1498–1501.

[5] A. Cerpa and D. Estrin, “ASCENT: adaptive self-configuring sensor networks topologies,” in *IEEE INFOCOM*, 2002, pp. 1278–1287.

[6] H. Zhang and J. C. Hou, “Maintaining sensing coverage and connectivity in large sensor networks,” *International Journal of Wireless Ad Hoc and Sensor Networks*, vol. 1, no. 1–2, pp. 89–124, 2005.

[7] Y. C. Wang, C. C. Hu, and Y. C. Tseng, “Efficient placement and dispatch of sensors in a wireless sensor network,” *IEEE Transactions on Mobile Computing*, vol. 7, no. 2, pp. 262–274, 2008.

[8] Y. C. Wang and Y. C. Tseng, “Distributed deployment schemes for mobile wireless sensor networks to ensure multilevel coverage,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 9, pp. 1280–1294, 2008.

[9] M. S. Pan, H. W. Fang, Y. C. Liu, and Y. C. Tseng, “Address assignment and routing schemes for ZigBee-based long-thin wireless sensor networks,” in *IEEE Vehicular Technology Conference*, 2008, pp. 173–177.

[10] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed diffusion: a scalable and robust communication paradigm for sensor networks,” in *ACM International Conference on Mobile Computing and Networking*, 2000, pp. 56–67.

[11] B. Krishnamachari, D. Estrin, and S. Wicker, “The impact of data aggregation in wireless sensor networks,” in *IEEE International Conference on Distributed Computing Systems Workshops*, 2002, pp. 575–578.

[12] A. F. Harris III, R. Kravets, and I. Gupta, “Building trees based on aggregation efficiency in sensor networks,” *ACM Ad Hoc Networks*, vol. 5, no. 8, pp. 1317–1328, 2007.

[13] M. J. Handy, M. Haase, and D. Timmermann, “Low energy adaptive clustering hierarchy with deterministic cluster-head selection,” in *IEEE International Workshop on Mobile and Wireless Communications Network*, 2002, pp. 368–372.

[14] S. Lindsey and C. S. Raghavendra, “PEGASIS: power-efficient gathering in sensor information systems,” in *IEEE Aerospace Conference*, 2002, pp. 1125–1130.

[15] O. Younis and S. Fahmy, “Distributed clustering in ad-hoc sensor networks: a hybrid, energy-efficient approach,” in *IEEE INFOCOM*, 2004, pp. 629–640.

[16] Y. Zhu, R. Vedantham, S. J. Park, and R. Sivakumar, “A scalable correlation aware aggregation strategy forwireless sensor networks,” in *IEEE International Conference on Wireless Internet*, 2005, pp. 122–129.

[17] Jennic microcontrollers. [Online]. Available: http://www.jennic.com/

[18] LAN/MAN Standards Committee of the IEEE Computer Society, “IEEE Std 802.15.4-2003, Wireless medium access control (MAC) and physical layer (PHY) specifications for low-rate wireless personal area networks (LR-WPANs),” *IEEE*, 2003.